Toward Efficient Learning of Structured Representations in Computer Vision

Phuc H. Le Khac B.Sc.

Supervised by Prof. Alan F. Smeaton & Dr. Graham Healy
Dr. Derek Greene, University College Dublin

A thesis presented for the degree of Doctor of Philosophy (PhD)

School of Computing
Dublin City University

August 2024

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Phuc H. Le Khac, ID No.: 19214311, Date: 26-08-2024

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Professor Alan Smeaton and Dr. Graham Healy, for their invaluable guidance and unwavering support throughout this research journey. Their expertise and encouragement have been instrumental in shaping this work. This would not have been possible without them. I am also grateful to my external supervisor, Dr. Derek Green, for his insightful contributions and steadfast support.

A special thanks goes to Angela Lally and ML-Labs CRT managers for their constant assistance and support throughout the entire process. This help has been truly invaluable.

I would like to pay tribute to Dr. Kevin McGuinness, whose impact on my work and personal growth cannot be overstated. Our interactions, though brief, were profoundly influential and will always be cherished.

I am deeply grateful to Science Foundation Ireland for their generous funding support, which made this research possible. My sincere appreciation also extends to Dr. Christian Burnham, Dr. Romain Bregier, and Dr. Vincent Leroy for sharing their experiences and providing invaluable guidance during my internships. Their mentorship has significantly enriched my academic journey and contributed to the depth of this work.

Finally, I wish to express my heartfelt appreciation to my parents, family, and friends. Their unwavering trust, patience, and support have been my anchor throughout this challenging but rewarding journey. This accomplishment would not have been possible without their love and encouragement.

List of Publications

- **Le-Khac, Phuc H.**, Graham Healy, and Alan F. Smeaton. "Efficient Object-centric Representation Learning with pretrained geometric prior." (In submission).
- **Le-Khac, Phuc H.**, Vincent Leroy, and Romain Bregier. "Structured Representation Learning of Human Images using Latent Alignment." (In submission)
- **Le-Khac, Phuc H.**, Graham Healy, and Alan F. Smeaton. "Contrastive representation learning: A framework and review." IEEE ACCESS, 8 (2020): 193907-193934.
- **Le-Khac, Phuc H.**, Ayush K. Rai, Graham Healy, Alan F. Smeaton, and Noel E. O'Connor. "Investigating memorability of dynamic media." arXiv preprint arXiv:2012.15641 (2020).
- Cuong, Dinh Viet, **Phuc H. Le-Khac**, Adam Stapleton, Elke Eichlemann, Mark Roantree, and Alan F. Smeaton. "Managing Large Dataset Gaps in Urban Air Quality Prediction: DCU-Insight-AQ at MediaEval 2022." arXiv preprint arXiv:2212.10273 (2022).
- Dayma, Boris, Suraj Patil, Pedro Cuenca, Khalid Saifullah, Tanishq Abraham, **Phuc H. Le Khac**, Luke Melas, and Ritobrata Ghosh. "Dall· e mini.". Online: https://github.com/borisdayma/dalle-mini (accessed Sep. 29, 2022). DOI: 10.5281/zenodo.5146400 (2021).

Contents

List of Abbreviations

ACT Augmented Temporal Contrast.

AE AutoEncoder.

AI Artificial Intelligence.

ANN Artificial Neural Network.

ANN Artificial Neural Networks.

API Application Programming Interface.

AVE-Net Audio-Visual Embedding Network.

BERT Bidirectional Encoder Representations from Transformers.

BYOL Bootstrap Your Own Latent.

CDR Contrastive Representation Distillation.

CLIP Contrastive Language Image Pre-training.

CMA Cross-modal Agreement.

CMC Contrastive Multiview Coding.

CNN Convolutional Neural Network.

CPC Contrastive Predictive Coding.

CRD Contrastive Representation Distillation.

CRL Contrastive Representation Learning.

CroCo Cross-view Completion.

Cross-AVID Cross-modal Audio Visual Instance Discrimination.

DINO Self-**di**stillation with **no** label.

DL Deep Learning.

DPC Dense Predictive Coding.

EBM Energy-based Model.

GAN Generative Adversarial Network.

GIM Greedy InfoMax.

GPT Generative Pre-trained Transformer.

GPU Graphical Processing Unit.

iBOT Image BERT Pre-Training with Online Tokenizer.

LLM Large Language Models.

LN Layer Normalisation.

LoCo Local Contrastive.

MAE Masked-AutoEncoder.

MLP Multi-Layer Perceptron.

MOVi Multi-Object Video.

MSN Masked Siamese Networks.

NERF Neural Radiance Field.

NLP Natural Language Processing.

NT-Xent Normalised Temperature Cross Entropy.

OCRL Object-centric Representation Learning.

PCL Prototypical Contrastive Learning.

ReLU Rectified Linear Unit.

SAM Segment Anything Model.

SAVI Conditional Object-centric Learning on Video.

ST-DIM SpatioTemporal DeepInfoMax (ST-DIM).

SwAV Swapping Assignments between multiple Views of the same image.

TPU Tensor Processing Units.

VAE Variational AutoEncoder.

VDIM Video DeepInfoMax (ST-DIM).

VINCE Video Noise Contrastive Estimation.

ViT Vision Transformer.

VQ-VAE Vector Quantised-Variational AutoEncoder.

List of Tables

List of Figures

Toward efficient learning of structured representations in computer vision

Phuc H. Le Khac

Abstract

The ability to learn a hierarchical and compact representation from data stands as a fundamental principle behind the rapid growth of Deep Learning, particularly evident in Computer Vision. Despite the significant progress on the perception tasks such as recognition and detection, these models still fall short in terms of reasoning and planning capabilities, and cannot generalise systematically despite being trained with extensive amount of data and compute resources.

How to effectively scale up a representation learning system in terms of computation and data, and extend the capabilities of the visual representations toward high-level tasks is the central research topic of this thesis.

First we focus on contrastive representation learning, a general approach for learning representation by comparison. We survey and analyse more than 100 recent works and provide a framework to categorise and understand research in this direction, not only in the context of self-supervised visual learning but also for other domains and applications.

We then turn towards the problem of object-centric representation learning, a promising approach to learn structured representations in a complex visual scene for planning and reasoning tasks. We first explore using discrete representation for object-centric learning, motivated by the common goal of decomposing the continuous visual signal into individual discrete components.

Understanding the importance and challenges of scaling in learning representations from data, we propose an efficient architecture for decoding object-centric representations, a ubiquitous but memory-intense component present in most object-centric learning methods.

Finally, to address the challenge of learning these object-centric representations in complex and realistic data, we capitalise on the advancements in pre-trained models for visual representations, enabling the learning of higher-level representations. Inspired by human cognitive development, we further study the effects of depth information and geometry contained in these representations, exploring their influence on the process of unsupervised object discovery.

Chapter 1

Introduction

Artificial Intelligence (AI) broadly refers to the study of replicating the intelligent behaviours of humans and animals outside of biological systems. While the term AI was first coined in the 1950s to focus the study on digital computers, the idea of building an intelligent mechanical system dates back thousands of years and has been explored throughout history in art, novels and science fictions. The most recent form of AI that has entered the general public discourse is Generative AI, characterised by systems such as Large Language Models (LLM) and image generation system that can "talk" and "draw" with an unprecedented level of realism.

The driving component behind these recent successes is a technique called Deep Learning (DL), a sub-field of Machine Learning that emphasises training large Artificial Neural Networks (ANN) on large scale data and using an enormous amount of compute resources to do so. Deep neural networks are unique in their ability to learn a series of transformations on input data and to represent that data in an algebraic form such that these representations can be used for various downstream tasks. This is particularly helpful for solving tasks that deal with perceptual input data such as images, videos, audio and text.

As a subfield of Artificial Intelligence, Computer vision (CV) focuses on the study of enabling computers to understand and interpret visual data from the world around us. Deep Learning in general and Representation Learning in particular have played an integral role in the advancement and development of the field of Computer Vision. Thanks to the representational learning power of deep neural networks, programmers for visual tasks can avoid hardcoding brittle rules or heuristics to handle high-dimensional input data such as images and videos. Instead, they can leverage the learned representations in a lower-dimension vector space to perform those perception tasks.

Representation Learning is a sub-field of Deep Learning focusing on the topic of learning these representations. It involves the study of methods and techniques to make the learned representations become more general, powerful and efficient

such that they need less data to achieve higher accuracy on more downstream tasks. Representation Learning in Computer Vision is the principal topic of this thesis.

1.1 Hypotheses and Research Questions

In relation to the topic of representation learning in the visual domains, we consider the following two well-known hypotheses:

Scaling Hypothesis: As deep neural networks are scaled up in size and trained on more diverse data, they generalise better, become more sample-efficient for downstream tasks, and they exhibit emergent capabilities that they were not explicitly trained to do. This is often discussed in the context of the "Bitter Lesson" [264] which states that "The two methods that seem to scale arbitrarily in this way [with compute and data] are *search* and *learning*."

Structured Representation Hypothesis: Real world data is assumed to come from a *data generating process*. Capturing the underlying structure of this generative process within the representation space can enhance learning efficiency and facilitate generalisation in a systematic manner.

To advance our understanding and make progress in the field of Visual Representation Learning considering these two hypotheses, this thesis aims to address the following research questions:

Research Question 1: What architecture and training objective can help scaling up deep neural networks to learn a broadly useful representation of the visual world?

The empirical trend so far has been in support of the Scaling Hypothesis stated above, by showing that larger neural networks, trained on more diverse data with increased computational resources, yield representations that are more capable across various downstream tasks. However, these representations are often confined to their training distribution, lacking broad generalisation capabilities. Developing methods and systems for learning more general representations that serve as a foundational "commonsense" knowledge could be a pivotal advancement in both computer vision and broader AI.

Research Question 2: What are the general principles and helpful inductive biases in learning to enable such representation without task-specific labelled data?

The reliance on human-labelled data has been central to visual representation learning. As we move towards more massive neural networks, acquiring and curating

this expensive labelled data becomes increasingly challenging. Investigating the principles and inductive biases for learning without explicit human supervision is crucial in the transition towards self-supervised learning systems, a natural step forward in the direction of the Scaling Hypothesis.

Research Question 3: How can object-centric representation learning approaches, particularly slot-based methods, be designed to capture increasingly complex and abstract visual concepts in a structured manner?

Beyond the *content* encoded in the representation, exploring the *structure* of the representation itself remains relatively underexplored. Extending current methods to learn representation spaces that not only capture information but also encapsulate the underlying structure of the data can open up new possibilities for deep neural networks in diverse problem domains.

Research Question 4: What techniques can improve the computational efficiency of object-centric representation learning methods to enable scaling up these structured representation learning approaches?

To capture the structure of the visual world, by imposing more structure in the representation space could naively compromise efficiency. As learning systems scale up and become more computationally intensive, optimising the efficiency of this computation becomes paramount. Identifying and designing methods that can scale up object-centric learning methods could be an important step towards harmonising the contentions between the Scaling Hypothesis and the Structured Representation Hypothesis.

1.2 Thesis Structure

The remainder of this thesis is structured as follows.

Chapter 2: Background This chapter provides foundational knowledge on the field and the topic of the thesis. It begins with an overview of Computer Vision and its relationship to the broader field of AI. Following that, it covers the evolution, advantages, and limitations of Artificial Neural Networks, fundamental components underlying the success of Machine Learning and Deep Learning. Additionally, specific architectural components relevant to the later part of the thesis are discussed. The chapter then shifts focus to Representation Learning, a key factor in the success of deep learning. Lastly, the grand picture of a generally useful and capable AI system is presented as motivation for learning better representations, along with challenges hindering its realisation from the current state.

Chapter 3: Contrastive Representation Learning: A Framework and Review This chapter presents a framework and review for Contrastive Representation Learning, addressing Research Question 1 regarding building more powerful representation learning systems. Additionally, a comprehensive review of various methods and general principles in learning visual representations is provided, addressing Research Question 2 that examines understanding the principles in designing self-supervised learning systems.

For the remainder of the thesis, we turn our attention to the challenge of Object-centric representation, a topic that addresses learning structured representations, which is the focus of Research Questions 3 and 4.

Chapter 4: Overview of Object-centric Representation Learning This chapter offers a concise overview of Object-centric Representation Learning, including its motivation, goals, and developmental history and related work, bringing us to the current state of the art. The general framework for object-centric learning, foundational for subsequent experiments, is introduced.

From Chapter 5 to Chapter 7 we presents a series of experiments exploring how to improve different aspects of object-centric learning methods.

Chapter 5: Learning Discrete Object-centric Representations In this chapter, we present work on learning discrete object-centric representations. This novel approach replaces a continuous representation with a discrete representation, utilising the vector quantisation approach.

Chapter 6: Improving Efficiency in Object-centric Learning This chapter focuses on enhancing the efficiency of reconstruction-based object-centric learning methods. We introduce a simple attention mechanism in the object decoder components, leading to improved efficiency, reduced memory requirements, and lower compute demands.

Chapter 7: Unsupervised Object Discovery with Geometric Representation Investigating the problem of unsupervised object discovery through its learning signal, this chapter leverages pre-trained models with a specific focus on 3D geometry for object-centric representation.

Chapter 8: Conclusions We wrap up the topic of Contrastive Representation Learning presented in Chapter 3 and Object-centric representation learning presented from Chapter 4 to Chapter 7 reviewing recent progress and we appraise the topic of object-centric representation learning. This concluding chapter provides a

summary of our contributions, recaps the diverse topics in representation learning, summarise our answers to our 4 research questions and speculates on the future direction of the field.

Chapter 2

Background

In this chapter, a brief overview of the history and development of Machine Learning (ML) and Computer Vision (CV) in the broader context of Artificial Intelligence (AI) is provided. We particularly focus on the topic of Representation Learning from the visual domain. For a more detailed exposition on these topics and more, readers are advised to consult Goodfellow, Bengio, and Courville [86], Murphy [196], Murphy [195].

2.1 Computer Vision

The field of artificial intelligence (AI) aims to create intelligent machines that can mimic human cognitive abilities, including perception, recognition, reasoning, and decision-making. While the idea of mechanical robots and artificial intelligence can be traced back to as far as ancient Greek mythology [182], modern AI has its roots in the development of computers and computer science in the mid-twentieth century.

The first usage of the term "Artificial Inteligence" was in an ambitious proposal for a summer school by McCarthy et al. [183] in 1958 to research on "how to make machines use language, form abstractions and concepts, solve the kinds of problems now reserved for humans, and improve themselves". Since then, AI has grown to encompass a wide range of subfields and applications. Very broadly categorised, research in AI mostly focuses on reasoning and planning, which are based on logic approaches such as expert systems, or perceiving and understanding, which are based on learning approaches such as neural networks [232].

Computer vision (CV) is a subfield of Artificial Intelligence that focuses on the study of enabling computers to understand and interpret visual data from the world around us. Endowing computers with capabilities similar to those of the human visual system is an outstanding goal that is almost as old as the modern computer itself with references to work in the area going back more than 50 years such as Papert [209]. However, it was quickly realised that CV is a very challenging field of

research, as computer systems needed to be taught to interpret visual data in much the same way that humans do unconsciously, which is a complex and multifaceted process.

Early approaches to computer vision in the 1990s and 2000s revolved around extracting low-level visual features such as edges, corners, and textures from images and then using these features to identify and classify objects. These classical computer vision techniques, which include methods such as edge detection, histogram of oriented gradients and optical flow, are still used today, especially in real-time applications where processing speed is of the essence.

In the past decade, there has been a resurgence of interest in deep learning (DL) methods, which have revolutionised the field of computer vision (see [237, 235]). Deep learning algorithms, which are based on artificial neural networks, are particularly well-suited to tasks such as image classification, object detection, and segmentation, which have traditionally been challenging for classical computer vision methods based on the handcrafted features used previously [157]. Together with the increases in computing power and the availability of data, researchers have been able to train highly accurate computer vision systems based on deep neural network that can perform tasks once out of reach by previous methods.

While machine learning underlines the progresses of computer vision, computer vision has also played a crucial role in the development of many machine learning techniques which have subsequently been applied in other areas. For example, many of the fundamental building blocks such as Convolutional Neural Network [79, 161] and Residual Connection [108] were first developed in the context of computer vision for use in object segmentation, detection, and classification tasks, before being applied to other fields such as natural language processing and speech recognition. Many of the biggest improvement in NLP tasks like language translation, sentiment analysis, text summarisation, speech recognition, image and video captioning have all been enabled by deep learning techniques trickled down from CV researches [140]. More generally, thanks to Deep Learning's capabilities of handling both continuous and discrete signals, the research directions for many subfields of AI have been on a steady convergent trajectory.

Computer vision has a wide range of application areas, from medical imaging to autonomous vehicles, social media, and robotics. In medical imaging, for example, computer vision techniques are used to help doctors diagnose diseases [56], read X-Ray images [279], screen Computed Tomography scans [8], identify tumours [62], and monitor patient health [147] . In autonomous vehicles, computer vision systems are used to detect and track other vehicles, pedestrians, and obstacles on the road, and to make real-time decisions about how to navigate safely [129]. In social media, computer vision algorithms can be used to automatically tag and organise photos

and videos which have been uploaded and shared on social media platforms [3], or used in animated emojis and cosmetic filters [257]. In robotics, they can help robots navigate complex environments [59] and manipulate objects with greater precision and accuracy [5].

Even with the tremendous progresses achieved, nowadays computer vision is still a very active research topic that holds the promise to enable even more complex and capable vision-based applications. One notable recent example is the release of the Segment Anything Model (SAM) [152] by Meta AI Research in April 2023. SAM combines advances in large scale training with a lightweight, prompt-able decoder architecture, enabling "model-in-the-loop" data engines that scale up to more than 1 billion masks in 11 million images. Thanks to its open-source code, models and data, in the span of a few weeks during Spring of 2023 there have been numerous subsequent modifications and extensions such as in-painting, tracking and video segmentation[1].

2.2 Artificial Neural Networks

The Artificial Neural Network is at the very heart of the Machine Learning and Deep Learning revolution of the past decade. As a field, Machine Learning and Deep Learning have a huge influence and impact on the discipline of computer science, such that sometimes it is referred to as "Software 2.0" to mark a major paradigm shift in the way computer programs will be written [143].

2.2.1 Machine Learning

Machine Learning, as the name suggests, revolves around the idea of machines and computers acquiring knowledge and improving their performance in some prediction or classification task based on experience and data. Broadly speaking, Machine Learning is the study of designing an algorithm, or statistical model, f, that can adapt and adjust its output based on a data pairing of input and output $\mathbf{X}, y$: $\hat{y} = f_\theta(\mathbf{x})$. The crucial difference with a classical algorithm is its ability to make predictions without having to be explicitly programmed. This make it particularly promising for problems where the input domain is high-dimensional and is very hard or impossible to explicitly enumerate all the rules. This fundamental concept underpins the entire field of ML and distinguishes it from traditional, rule-based programming.

At the core of machine learning is the utilisation of data as the primary source of knowledge. ML algorithms are fed large datasets containing relevant information,

[1] https://github.com/Hedlen/awesome-segment-anything

such as images, text, and audio, represented as numerical values. By analysing and processing this data, ML algorithms discover patterns, correlations, and underlying structures, and can make categorisations or predictions on similar, new and unseen data points.

The behaviour of such a "machine" is governed by its parameters, denoted θ. Machine learning models undergo a training phase where they learn from training examples. During this phase, the program "learns" by iteratively adjusting its internal parameters to minimise the differences between its predictions and the true outcomes in the training data.

Once trained, the model's ultimate goal is to generalise its knowledge to make accurate predictions, or inferences, on unseen, unlabelled data. Generalisation is a hallmark of learning, as it indicates the model's ability to apply its acquired knowledge to new situations, as opposed to just memorising its training samples.

Learning is an optimisation process and this process is often guided by an objective function, like minimising prediction errors or maximising rewards in reinforcement learning. The optimisation objective for learning is often referred to as the "loss function", or "reward function". The gradient of the objective with respect to the models' parameters are computed and this is used by various optimisation algorithms to minimise the loss, or to maximise the reward.

The concept of machine learning can be traced back to the 1940s when the idea of algorithms and models that could learn from data was introduced. One of the key developments in the field of machine learning was the creation of Artificial Neural Network (ANN), which was inspired by the structure of connections of biological neurons in the brain. The first artificial neural network, known as the McCulloch-Pitts neuron, was introduced by Warren McCulloch and Walter Pitts in 1943 [184]. This marked a significant milestone in the development of artificial intelligence and laid the foundation for subsequent neural network research. The McCulloch-Pitts neuron took binary inputs and applied a set of logical rules to produce binary outputs. It could perform basic logical operations like AND, OR, and NOT. While the McCulloch-Pitts neuron was a crucial theoretical development, it had limitations. It was not a learning algorithm since it could not adapt or learn from data, and its functionality was confined to specific pre-defined logic functions. Despite its simplicity, this early work enabled the development of more complex artificial neural networks.

The connection to the biological world has played a significant role in shaping the evolution of neural networks and their applications. The next significant step in the history of artificial neural networks was the invention of the Perceptron, developed by Frank Rosenblatt in 1957 [229], that could learn to recognise simple patterns in data. The perceptron was designed to be a binary linear classifier, capable of distinguishing

between two classes of data. It incorporated weighted inputs, a summing function, and a threshold activation function. One of the key innovations of the perceptron was its learning algorithm. It could automatically adjust the weights assigned to each input based on the success or failure of classification. This made it the first machine learning model capable of learning from data.

Soon it became apparent that the perceptron also had many limitations. It could only solve linearly separable problems and it was incapable of handling more complex, non-linearly separable data. Due to its limited capabilities, after its initial introduction, the perceptron in particular and neural networks in general received a lot of doubt and criticism about their potential usage [189].

Nonetheless, the development of the perceptron was a crucial step in the history of artificial neural networks. It demonstrated the potential of learning algorithms and laid the groundwork for the resurgence of interest in neural networks in later decades, especially with the advent of deep learning and more sophisticated neural architectures.

2.2.2 Deep Learning

Deep learning [235, 237] represents a transformative extension of traditional machine learning techniques, offering more powerful tools for data analysis, pattern recognition, and decision-making. It focuses on designing and training large, hierarchical neural networks composed of multiple layers, where the output of one layer is the input to another. This extension builds on the foundation of machine learning while introducing key innovations that enable the development of highly complex models capable of solving a wide range of tasks. Similar to how machine learning that took inspiration from the neuron, Deep Learning is also loosely inspired by the structure and function of the human brain, where information is processed through interconnected layers of neurons.

While a shallow neural network is usually just considered as one of many tools in machine learning, deep learning is synonymous with artificial neural networks and consists of multiple layers of interconnected artificial neurons. Each layer processes information and passes it to subsequent layers, creating a stack of transformations. This architecture allows deep networks to model increasingly complex and abstract representations of the input data [122].

One of the core principles of deep learning is its ability to automatically learn hierarchical feature representations from raw data. Traditional machine learning often relies on handcrafted features, requiring domain expertise and substantial effort to engineer relevant input features for a given problem. In contrast, deep learning models, such as deep neural networks, can automatically learn and extract

features at multiple levels of abstraction. This hierarchical representation enables deep learning models to capture intricate patterns and relationships in data without the need for explicit feature engineering.

As suggested by its name, the depth of a deep neural network is a key factor in their power. It allows these networks to capture complex patterns, dependencies, and hierarchies in data. As information passes through successive layers, the network can learn to represent high-level features that are composed of lower-level features [306].

Training deep neural networks is made possible by several critical factors, including innovations in optimisation algorithms, the power of parallel computing, and the abundant availability of large labelled datasets. These elements work in synergy together to enable deep learning models to efficiently tackle the immense complexity of some kinds of real-world data. These critical factors are discussed below:

Availability of Large Datasets: A crucial factor contributing to the success of deep learning is the availability of extensive and labelled datasets. In the past, collecting and annotating such datasets was a significant bottleneck to progress. However, recent years have seen an explosion in the collection and sharing of data, facilitated by the growth of the internet and advances in data storage and processing. Datasets like ImageNet [58], COCO [164], and various medical image collections contain millions of labelled examples, providing the necessary diversity and volume for training sophisticated deep models. These datasets enable models to discern complex patterns and narrow the divide when it comes to generalising to new, unseen data.

Parallel Computing: The training of deep neural networks requires substantial computational power, and this demand is met through parallel computing resources. Graphical Processing Unit (GPU) were initially developed to meet the demanding problem of rendering many different video pixels in real time, mostly in gaming applications. Once this parallel processing power was applied to Deep Learning[48], it enabled for the first time the ability to learn from large amounts of data. The culmination of GPU parallel processing and larger datasets resulted in the famous "ImageNet" moment in computer vision [157]. Realising the importance of specialised, parallel hardware, many different accelerators such as the Tensor Processing Units (TPU) together with many improvement on traditional GPUs have since been developed. These are key components that have played a pivotal role in driving the progress of Deep Learning in recent years. These specialised hardware accelerators can handle the matrix and vector operations inherent in neural network training with remarkable speed and efficiency. The parallel processing capabilities of GPUs

and TPUs enable the simultaneous computation of numerous model updates, dramatically reducing the time required to train deep neural networks.

Optimisation Techniques: Techniques like backpropagation [166] [231] for computing the gradient of parameters through many layers, stochastic gradient descent (SGD) and more advanced optimisers enable the efficient adjustment of even billions of model parameters.

In unison, optimisation algorithms, parallel computing, and large datasets enable deep learning models to handle an array of complex tasks, from image classification and natural language understanding to reinforcement learning in game playing and robotics. These advances have ushered in a new era of AI and machine learning applications, with deep learning models being employed in various fields, including healthcare, autonomous vehicles, and finance. As the field continues to evolve, researchers explore innovative ways to optimise the training processes, make efficient use of computational resources, and work with ever-growing datasets, paving the way for even more remarkable achievements in the future with the use of deep learning.

Research efforts have also focused on developing techniques for network architecture search and optimisation, leading to the emergence of automated methods for designing neural network architectures. These advances, coupled with the rise of transfer learning and pre-trained models, have made it easier to apply neural networks to various tasks, even with limited labelled data.

While neural networks and deep learning have seen significant advances and widespread adoption in recent years, they also face several limitations and setbacks that hinder progress and further adoption in the real world.

Computational Power: Training extensive neural networks with numerous layers and billions of parameters demands substantial computational resources, often beyond reach due to their prohibitive costs. Furthermore, deploying state-of-the-art models for inference after training present additional challenges, particularly when operating within constrained computing budgets [60]. The ethos of open development and open-source initiatives remains integral to the machine learning domain. Nonetheless, deep learning research is becoming less and less open, partly due to the growing expenses associated with training and utilising large foundational models, potentially impeding its open development and advancement. An example is the growing literature involving the study of the closed-source GPT4 model [206], sometimes just to answer the question of whether the model has been updated by its parent company over time [198].

Reliance on Labelled Training Data : As the size and compute requirements of neural networks increases, its reliance on large amounts of labelled training data used to learn to make accurate predictions also increases. Data collection and annotation processes are time-consuming, expensive, and often prone to errors. Limited training data results in overfitting, where the network memorises the training examples instead of generalising from them, leading to poor performance on unseen data [153]. Addressing the challenges related to data collection and annotation, as well as finding strategies to mitigate the effects of overfitting, represents an ongoing endeavour within the deep learning community [152]. These efforts are essential for maintaining the effectiveness and reliability of increasingly complex neural networks, as well as for advancing the field's capability to work with diverse, real-world datasets.

Interpretability and Explainability: Neural networks are often considered black-box models, meaning that it is challenging to understand the reasoning behind their predictions or decisions. While the working principle of individual neurons are simple and easy to interpret, the emergent capabilities of large networks with multiple neurons distributed over multiple layers are considerably more challenging to measure and understand [29]. This lack of interpretability and explainability is a significant setback in domains where transparency and accountability are crucial, such as healthcare and finance.

Lack of Understanding of Network Architectures: Designing the architecture of neural networks was more of an art than a science during the early stages. There was only a limited understanding of the optimal number of layers, the number of neurons in each layer, and the connectivity patterns between the neurons [131].

The specific architecture of modern deep learning models usually takes inspiration from many different fields such as neural science, signal processing, statistical learning and even quantum mechanics. But ultimately, modern deep learning architectures are driven by empirical results.

2.3 Architectures

There exist many different architecture components in deep learning. Neural science is a rich source of motivations for many early designs such as the Perceptron [229] or the DropOut mechanism to combat overfitting [254]. Nowadays, deep learning architectures are generally divided and grouped into layers, where each specific layer has its own characteristics. Layers are then connected sequentially or in parallel to form a deep computation graph, which are broadly referred to as models.

Below, we briefly review some general deep learning layers and principles that are relevant to image processing and representation learning.

2.3.1 Feed-forward Layer and Multi-Layer Perceptron

Feed-forward layers and Multi-Layer Perceptrons (MLPs) represent fundamental building blocks in the domain of artificial neural networks. These structures are the backbone of deep learning, empowering neural networks to learn intricate patterns and make accurate predictions across a wide array of input data and tasks.

Feed-forward Layer A feed-forward layer, also known as a dense layer or fully connected layer, is the simplest and most common building block in neural networks. All nodes in the previous layers are densely connected to every node in the next layer, hence its name. It forms the core of many neural architectures, including MLPs. The primary function of a feed-forward layer is to transform its input data through a linear operation, followed by a non-linear activation function.

The architecture of a feed-forward layer can be described as follows:

- Input Neurons (Nodes): Each node in the input layer represents a feature or component of the data. The number of input nodes corresponds to the dimensionality of the data.

- Weights and Bias: Associated with each input node is a weight, which quantifies the importance of that input in the layer's computations. Additionally, there is a bias term that allows the layer to learn an offset.

- Affine Transformation: The layer performs a linear combination of the input values and weights, summing the products of inputs and weights along with the bias term. Mathematically, this is represented as: $z_j = \sum_{i=1}^{N}(x_i \cdot w_{ij}) + b_j$, where z_j is the j-th output of the affine transformation, x_i are the input values, w_{ij} are the corresponding weights, N is the number of input nodes, and b_j is the bias. This step is repeated many times for different values of weights and biases to form the set of outputs nodes and is often perform in parallel as a single matrix-vector multiplication.

- Activation Function: The output of the affine transformation ($\mathbf{z}$) is then passed through a non-linear activation function, such as the sigmoid, Rectified Linear Unit (ReLU), or hyperbolic tangent (tanh). This activation function introduces non-linearity into the model, allowing it to capture complex relationships within the data.

- Output: The result of the activation function serves as the output of the feed-forward layer, which can be passed to subsequent layers or used as the final output of the neural network.

Multi-Layer Perceptron An MLP, or multi-layer perceptron, extends the concept of feed-forward layers to create a network with multiple layers, enabling it to learn hierarchical and more complex representations of data. An MLP typically consists of an input layer, one or more hidden layers, and an output layer. Each hidden layer contains one or more feed-forward layers, and the activation functions within these layers can vary.

The architecture of an MLP is characterised by the following:

- Input Layer: The input layer receives the raw data and passes it to the subsequent hidden layers. Each node in the input layer corresponds to a feature of the input data.

- Hidden Layers: These intermediate layers, placed between the input and output layers, are composed of feed-forward layers with non-linear activation functions. The number of hidden layers and nodes in each layer can be adjusted to suit the complexity of the task.

- Output Layer: The final layer of the MLP produces the network's predictions. The architecture of this layer depends on the nature of the task, such as regression, classification, or other specific objectives.

- Forward Propagation: During forward propagation, data flows through the network and each layer applies the affine transformation and activation function to progressively transform more abstract and informative representations.

- Backward pass: To train the MLP, backpropagation and optimisation techniques like gradient descent are employed. Backpropagation computes the gradients of the loss function with respect to the model's parameters, from the output layers back to the input layer, allowing the network to adjust its weights and biases to minimise the loss.

The fully-connected layer and MLP are the bedrocks of an artificial neural network due to their simplicity, universality and expressiveness. However, it also makes it less efficient in learning from more structured data like images and videos.

2.3.2 Convolutional Layer and Convolutional Neural Network

Convolutional layers and Convolutional Neural Networks (CNNs) [161] [202] are a family of architectures that are particularly effective in analysing visual data. They are widely used in computer vision tasks, and serve as the backbone for many different models such as in image classification, object detection, and image segmentation.

Convolutional Layers: A convolutional layer is a fundamental building block in a CNN, designed to perform a specialised operation called a convolution. The convolutional layer is designed to capture local patterns and spatial hierarchies in the data. Convolutional layers are crucial for recognising patterns, edges, textures, and more in image data.

Their architecture can be described as follows:

- Convolution Operation: The convolution operation involves sliding a small filter (also known as a kernel) over the input data, typically an image. At each position, the filter computes the element-wise product between its weights and the corresponding section of the input data. These products are then summed to produce a single value at that position in the output, called a feature map. It can be described as: $(f * g)(x, y) = \sum_i \sum_j f(i, j) * g(x - i, y - j)$, where x, y are coordinates of the input and output while i, j are the coordinates for the filter.

- Shared Weights: One of the key features of convolutional layers is weight sharing. The same filter is applied at multiple positions across the input data. This property dramatically reduces the number of parameters in the model compared to fully connected layers, making convolutional layers highly efficient and capable of capturing local patterns.

- Stride: Stride determines how much the filter shifts (or slides) across the input data after each operation. A larger stride reduces the size of the output feature map and decreases the computational cost.

- Padding: Padding is the addition of zeros around the input data before applying the convolution. It helps maintain the spatial dimensions of the feature maps produced by the convolutional layers. Padding can be 'valid' (no padding) or ' 'same' (padding is computed to keep the output size the same as the input).

- Activation Function: After the convolution operation, an activation function, such as ReLU (Rectified Linear Unit), is applied element-wise to introduce

non-linearity into the model.

Convolutional Neural Networks (CNN): A Convolutional Neural Network is a deep learning model composed of multiple layers, including convolutional layers, pooling layers, and fully connected layers. It is named after its most important layer, the convolutional layer. In a convolutional neural network (CNN), a convolutional layer is responsible for extracting features from input data (or previous layers) using convolution operations.

The architecture of a CNN can be described as follows:

- Input Layer: The input layer receives the raw data, typically an image, and passes it through a series of convolutional layers to extract hierarchical features.

- Convolutional Layers: Convolutional layers, as described earlier, are responsible for detecting local patterns and features in the input data for each respective layer.

- Pooling Layers: After each set of convolutional layers, pooling layers are often introduced to reduce the spatial dimensions of the feature maps. Pooling layers aggregate information from small regions of the feature maps, reducing the computational burden and promoting translation invariance.

- Fully Connected Layers: Toward the end of the CNN architecture, one or more fully connected layers are employed. These layers take the high-level features extracted by the previous layers and use them to make the final predictions.

- Output Layer: The output layer of the CNN is responsible for producing the network's predictions. The architecture of this layer depends on the task, with classification tasks often using softmax activation for probability distributions over classes.

- Training: CNNs are trained through backpropagation and gradient descent to optimise the model's weights and biases. Large labelled datasets are typically required for training CNNs to achieve high accuracy.

Convolutional Neural Networks are foundational in computer vision, image processing, and pattern recognition, offering a structured and efficient way to model complex relationships in image data. In summary, they can be considered as a locally-connected with weight-sharing version of the fully-connected layer. This reduces the number of parameters and overfitting challenges of the MLP, but it still shares the principle of a feed-forward architecture.

2.3.3 Attentional Layer and Transformer Architecture

Attention layers and the Transformer architecture have revolutionised natural language processing and deep learning in general by enabling models to focus on specific parts of input sequences, effectively capturing long-range dependencies and improving the handling of sequential data. Here, we explore the architecture of attention layers and how they are integrated into the Transformer model.

Attention Layers: The attention mechanism in deep learning [14], as its name implies, is inspired by the idea of attention in the human brain. An attention layer is a fundamental component in many modern neural network architectures. It allows the model to assign different levels of importance to different elements in an input sequence, focusing on relevant information and ignoring irrelevant ones. Crucially, the weighting of input, or attention score, are computed dynamically based on the input or some additional source of data.

The architecture of an attention layer can be described as follows:

- Input Sequence: An attention layer receives an input sequence, typically a sequence of vectors or embeddings. Let's denote the input sequence as $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$, where n represents the number of elements in the sequence.
- Query, Key, and Value Matrices: To compute attention scores, the input sequence $\mathbf{X}$ is transformed into three matrices: the query matrix (Q), the key matrix (K), and the value matrix (V). These matrices are learned during training. Mathematically, we can represent this as: $\mathbf{Q} = \mathbf{X} \cdot \mathbf{W}_Q$, $\mathbf{K} = \mathbf{X} \cdot \mathbf{W}_K$, $\mathbf{V} = \mathbf{X} \cdot \mathbf{W}_V$. Here, $\mathbf{W}_Q$, $\mathbf{W}_K$ and $\mathbf{W}_V$ are learnable weight matrices.
- Attention Scores: The attention scores are calculated using the dot product between the query and key matrices, measuring the similarity between each query and each key. The scores are scaled for better stability and normalised using the softmax function to ensure they sum to 1: $\text{Attention}(Q, K) = \text{softmax}(\frac{\mathbf{Q} \cdot \mathbf{K^T}}{\sqrt{d_k}})$ where d_k is the dimension of the key vectors.
- The attention scores are used to compute a weighted sum of the value vectors, resulting in the output of the attention layer: $\text{Output} = \text{Attention}(\mathbf{Q}, \mathbf{K}) \cdot \mathbf{V}$. The output represents a refined representation of the input sequence, emphasising relevant elements based on the attention scores.

In the context of deep learning, the attention mechanism is often used in sequence modelling tasks, such as language translation or sentiment analysis. It allows the model to attend to different parts of the input sequence at different time steps, giving more weight to relevant words or phrases.

The attention mechanism represents a big step forward for Deep Learning as its dynamic computation of the attention scores break out from the traditional norm of simple fully connected or convolutional layers.

Transformer The Transformer architecture, introduced in the paper "Attention is All You Need" by Vaswani et al. [272], leverages attention layers to process sequential data efficiently. It has gained prominence in various natural language processing tasks, including machine translation, text generation, and more. Solutions to sequence learning tasks were previously dominated by Recurrent Neural Networks [120] at the time.

The architecture of the Transformer model can be summarised as follows:

- Input Embeddings: An input sequence is embedded into a set of vectors $X = [x_1, x_2, ..., x_n]$, where n is the sequence length. These embeddings can be learned during training or obtained from pre-trained models.

- Positional Encodings: Since the Transformer does not have built-in sequential information, positional encodings are added to the input embeddings to provide information about the order of elements in the sequence.

- Stack of Encoder Layers: The Transformer comprises a stack of identical encoder layers. Each encoder layer includes a multi-head self-attention mechanism followed by an MLP, which allows the model to focus on different parts of the input sequence simultaneously. The output of each encoder layer is fed into the subsequent layer.

- Multi-Head Self-Attention: In each encoder layer, the input embeddings are transformed into query (Q), key (K), and value (V) matrices, as described earlier in the attention layer section. Multi-head attention consists of multiple parallel self-attention mechanisms, allowing the model to capture different relationships within the input.

- Feed-Forward Neural Networks: After multi-head self-attention, a feed-forward neural network processes the output, adding non-linearity to the model.

- Layer Normalisation and Residual Connections: Layer normalisation and residual connections are applied to each sub-layer, aiding in the stability and training of the model.

- Stack of Decoder Layers: For tasks like machine translation, the model includes a stack of decoder layers after the encoder layers. The decoder layers consist of masked self-attention, which prevents each position from attending to future positions, and cross-attention, which attends to the output of the encoder.

- Output Layer: The final layer of the Transformer model produces the model's predictions for the given task.

The Transformer architecture's innovative use of attention layers has led to remarkable advances in the field of natural language processing, enabling models to handle sequential data more effectively and efficiently.

Vision Transformer The transformer has become a general multipurpose powerful architecture. In computer vision The Vision Transformer, or ViT, is an adaptation of the Transformer architecture for computer vision tasks, bringing the success of attention mechanisms from natural language processing to the world of image analysis. Introduced by Dosovitskiy et al. in the paper "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale" [67], the ViT challenges traditional convolutional neural networks (CNNs) by demonstrating that pure attention-based models can excel in image classification and other vision-related tasks. The crucial difference when compared to a vanilla Transformer architecture can be described as follow:

- Patch Embeddings: Unlike traditional CNNs that work with pixel-level data, the ViT divides an image into non-overlapping patches and linearly projects each patch into an embedding vector. These patch embeddings are treated as the input sequence for the Transformer, thus enabling the model to handle structured image data.

- Classification Head: After the sequence of transformer encoder layers, a classification head is added to the ViT to make predictions for the given vision task, such as image classification. The classification head typically includes a pooling layer to aggregate information from different patches before producing the final output.

One of the distinguishing features of the ViT is its ability to leverage attention mechanisms to capture both local and global contextual information within images and throughout the networks. By using attention, it can learn to focus on specific patches and features that are relevant for making accurate predictions, making it highly effective for a wide range of computer vision tasks.

The Vision Transformer has made a significant impact on the field of computer vision, offering an alternative to CNNs by demonstrating its potential for image recognition, object detection, and segmentation tasks. Its architecture has inspired various subsequent variations and research, emphasising the power of attention mechanisms in the realm of visual information processing.

Consolidation of Transformer Architecutre The consolidation of the Transformer architecture across various domains represents a remarkable paradigm shift in machine learning. Originally designed for natural language processing tasks, the Transformer has proven to be highly adaptable, demonstrating its versatility in handling diverse data modalities and applications. Its fundamental building blocks, including self-attention mechanisms and multi-head attention, have been repurposed and reconfigured to excel in domains beyond text processing. These range from computer vision, where the Vision Transformer (ViT) has achieved impressive results in image analysis, to audio processing with models like the Audio Spectrogram Transformer [85], and even reinforcement learning in the form of the Decision Transformer [39] and others. The Transformer's impact has transcended domain and modality boundaries. Its modular and scalable architecture, coupled with its ability to capture complex relationships within data, has catalysed a unifying trend in AI research. The "Flash Transformer" [53] for instance, has pioneered novel ways to enhance the speed and efficiency of Transformer models and becomes instantly applicable to a wide range of domains, thanks to the ubiquity of the Transformer architecture.

This consolidation signifies a cross-pollination of ideas and techniques, fostering innovation and opening up new frontiers in machine learning and artificial intelligence, with the Transformer architecture at its core.

2.4 Representation Learning

Deep Learning has becomes an essential building block of any system that learns from high-dimensional, unstructured data such as images, videos, text or audio. In the early day of Machine Learning, much research effort was spent on designing data transformation and pre-processing pipelines, and learning was only used to make a shallow decision based on these hand-crafted features. One of the key ingredients in the success of deep learning is its ability to automatically learn and extract through deep layers, some useful features from the data.

The increase in available computation and datasets has enabled the paradigm shift from using hand-designed feature extractors to learned feature extractors. As a result, the focus in research also shifted from feature-engineering to architecture-engineering. Research into deep learning architectures has exploded in recent years and has matured into a few core principles and building blocks e.g convolution layers [161] for spatial data, recurrent layers [120] for sequential data, and attention layers [14] for set data. By stacking these building blocks into deep networks that can be optimised end-to-end, these models can learn a hierarchical, distributed, compact yet expressive representation of its input data.

The representation mappings learned by deep neural networks offer several ben-

efits. By virtue of compression, the representations tend to ignore unimportant variations in the data and can generalise to unseen inputs. Additionally, these representations can be learned and optimised directly for a downstream task of interest, while possessing remarkable transferability to other tasks with similar input domains [302]. Finally, it is also possible to learn a powerful and general representation space for a given input domain, that is applicable to a wide range of tasks, in some cases with better performance than when optimising for a downstream task directly [41]. This approach is the primary motivation behind deep representation learning, which continues to be a focus of research and development in deep learning and machine learning in general.

2.4.1 What is Representation Learning ?

Representation learning refers to the process of learning a parametric mapping from the raw input data domain to a feature vector or tensor, with the aim of capturing and extracting more abstract and useful concepts that can be used for a range of downstream tasks. The performance of a machine learning system can be measured using several metrics including efficiency in the training process, accuracy of its output and overall effectiveness, and this is directly determined by the choice and quality of the data representation, or features, in the data used to train it. While it is obvious that some criteria for usefulness depend on the task, it is also universally assumed that there are sets of features that are representative of a dataset and that are generally useful as input for many kinds of downstream classifier or predictor. Focusing explicitly on learning representation in some cases can be beneficial, for example, when a labelled dataset for a task is small and we want to leverage a larger unlabelled dataset to improve the performance of a learning system.

Often the input domain is high-dimensional and even multi-modal (images, video, sound, text) and the encoded feature is represented in a manifold of a much lower dimensionality. While all dimensionality reduction methods convert high-dimensional inputs to a lower-dimensional representation, not all methods learn a mapping that meaningfully generalises on new data samples, and that is what representation learning does.

2.4.2 What Makes a Good Representation ?

As a goal, the task of explicitly learning a good representation for data in comparison to implicitly learning a good representation to optimise performance for a task, can be tricky. Firstly, it is not entirely clear what makes a good representation. Based on the analysis by Bengio, Courville, and Vincent [21], a good representation is locally smooth in its manifold, is temporally and spatially coherent in a sequence

of observations, has multiple, hierarchically-organised explanatory factors which are shared across tasks, has simple dependencies among factors and is sparsely activated for a specific input.

The field of deep representation learning has developed a number of core principals in learning good representations:

- **Distributed**: Representations that are expressive and can represent an exponential amount of configurations for their size. This is in contrast to other types of representations such as one-hot encoding, as learned by many clustering algorithms;
- **Abstraction and Invariant**: Good representations can capture more abstract concepts that are invariant to small and local changes in input data;
- **Disentangled**: While a good representation should capture as many factors and discard as little data as possible, each factor should be as disentangled as possible. Aside from promoting feature reuse in learning systems, it can also be beneficial for other purposes such as explainability;
- **Composable and transferable**: To maximise its usefulness, representations should be able to compose in order to form novel concepts and to transfer its learned knowledge to many different tasks. This property will enable chaining different neural networks together, or combining and performing arithmetic on the representation space.

These principles describe an ideal representation that can express a large number of input configurations for their size, capture more abstract concepts that are invariant to small and local changes in data, compose features for different purposes, and transfer knowledge to many different tasks.

2.4.3 How to Learn Representations

Learning good representations of data is a complex task with many different methods and algorithms available. One way to categorise the available approaches is by considering two key axes of learning: generative versus discriminative modelling, and supervised versus unsupervised learning. By examining each of these dimensions, we can gain a better understanding of the various techniques available and their respective strengths and weaknesses.

Generative and discriminative modelling

In the machine learning literature, approaches to learning representations of data are often divided into two main categories: generative or discriminative modelling.

While both approaches assume that a good representation will capture the underlying factors that explain the inputs, they differ in the process of learning to model the data.

From the perspective of statistical learning, the representation is also called the unobserved latent variable, and the process of inferring the latent representation from data is called **inference**.

Generative approaches learn representations by modelling the data distribution $p(\mathbf{x})$, for example: all the pixels in an image. They are based on the assumption that a good model $p(\mathbf{x})$ that can generate realistic data samples, must also in turn capture the underlying structure related to the explanatory variables $\mathbf{y}$. Evaluating the conditional distribution $p(\mathbf{y}|\mathbf{x})$ for some discriminative tasks on variable $\mathbf{y}$ can then be obtained by the application of Bayes' rule.

Discriminative approaches to learning representations on the other hand learn representations by directly modelling the conditional distribution $p(\mathbf{y}|\mathbf{x})$ with a parametrised model that takes as input the data sample $\mathbf{x}$ and outputs the label variable $\mathbf{y}$. Discriminative modelling consists of an inference step that infers the values of the latent variables $p(\mathbf{v}|\mathbf{x})$, and then directly makes downstream decisions from those inferred variables $p(\mathbf{y}|\mathbf{v})$.

Discriminative models have some advantages when compared to generative models. Modelling the distribution for the set of data is computationally expensive and is not necessary in order to extract representations. If the goal is only to learn a mapping to a lower dimension representation, the generation process in a generative model can be considered wasteful. In addition, the task of learning a good decoder can be entangled with the task of learning a good feature encoder. The objective functions in generative models are also harder to design and more expensive to evaluate since they usually operate in the high-dimensional input space.

While there is no clear winner between generative and discriminative modelling, both have their advantages and can complement each other. Generative models are minimalist in terms of their training objective since they only need to generate or reconstruct the input data. Therefore, research in generative modelling is more focused on the inductive biases and architectures that facilitate learning a useful representation. On the other hand, discriminative models require careful design decisions for their objective training, which can sometimes be more important than the model architecture. Combining the strengths of both approaches is an important research direction, as it could potentially lead to more effective representation learning.

Supervised and Unsupervised Learning

In deep learning, supervised learning methods have traditionally been the most successful, where a representation is learned by mapping from input data to a corresponding human-generated label. Earlier paradigms involving pre-training layer-wise unsupervised models provided little or no benefit in the more modern end-to-end supervised setting. As the performance of deep learning can scale upwards with the amount of data and the model size [141], the need for labelled data has been identified as an impeding factor in scaling deep networks. However, the need for labelled data can be a limiting factor in scaling deep networks, as labelling data can be time-consuming, expensive, and potentially biased through the labelling process.

Until recently, most discriminative approaches to learning representations have been a type of supervised learning. Unsupervised representation learning methods, such as generative models, have previously been explored but are computationally expensive and that limited their ability to model dependencies between input dimensions. Some newer works under the term "self-supervised" learning aim to learn useful representations without labels using discriminative modelling approaches. These methods have shown great success when used for transfer learning, surpassing supervised pre-trained models in multiple downstream tasks, in both computer vision and natural language processing applications. Since a self-supervised discriminative model does not have human-generated labels corresponding to the inputs like its supervised counterparts, the success of self-supervised methods comes from the elegant design of the pretext tasks to generate a pseudo-label from part of the input data itself.

2.4.4 How to Evaluate Learned Representations

Evaluating the quality of a learned representation is not as straightforward as in supervised learning where we can directly optimise for a specific goal. Due to the flexible nature of learned representations, evaluating a good representation requires assessing its training objectives and metrics, as well as evaluating its usefulness, transferability, and generality across a range of downstream tasks.

In the self-supervised setting where representation is learned through a proxy task, the optimisation objective of the proxy task can serve as a proxy performance measure for the learned representation. For instance, for contrastive learning methods where the model is trained to bring similar representations closer and push dissimilar ones apart, the effectiveness of the learned representation can be gauged by how well it performs on this contrastive task, often measured by the alignment and uniformity of the feature distribution. Another example is the use of cross-entropy in the next-token prediction objective as the foundation of the scaling law, which are

used to predict the performance of LLMs as they are scaled up in size, training data, and compute resources. Representations learned by established models can serve as valuable benchmarks for evaluating other models, providing a means to assess the quality and similarity of learned features. One method for this is Centered Kernel Alignment (CKA), which measures the similarity between the representations of two models. By comparing a new model's representations to those of a well-known model, such as a pre-trained neural network, researchers can quantify how closely the new model's internal features align with established ones, offering insights into the model's generalisation and transferability. Similarly, in the evaluation of generative models, the Inception Score (IS) leverages the Inception model, a widely-used image classifier, to assess the quality of generated images. The IS calculates how well the generated images match the distribution of real-world images as recognised by the Inception model, thus providing a measure of both the fidelity and diversity of the generated samples. These approaches demonstrate how leveraging representations from established models can provide a robust and interpretable framework for evaluating new models, especially when direct evaluation may be challenging or when specific benchmarks are needed.

Sometimes, a good representation is valuable for studying the underlying characteristics of the data, even without the need for a particular task. For example, in unsupervised learning, a representation that clusters similar representations together can reveal the intrinsic structure of the dataset, aiding in exploratory data analysis. In a scientific research context for example, representations learned from biological data can reveal patterns or clusters that correspond to different biological processes or disease states, providing insights that are valuable beyond specific predictive tasks.

Ultimately, a good representation of data is determined by its performance on downstream tasks. For example, in the case of image classification, a robust representation will only need to capture the essential features of objects, such as shapes, textures, and colours, allowing the model to distinguish between different classes effectively. Similarly for detection, the representation must be rich enough to localise objects within an image, requiring certain spatial and contextual understanding. In segmentation, a good representation must not only identify objects but also delineate their boundaries at the pixel level. A good visual representation that captures the essential properties of the data and can be shared across these tasks. However, representations for these tasks can sometimes conflict, especially when the model's capacity is insufficient, resulting in a trade-off between performance in one task and generalisation across multiple tasks. One widely used approach to quickly evaluate representation on a wide range of downstream tasks is linear probing, where a simple linear classifier is trained on top of the frozen representations to assess how

well they capture the relevant features for downstream tasks. The performance of those linear classifier serves as an indicator of the representation's quality, where high accuracy suggests that the features learned by the model are well-organised and informative. Another method lightweight method is k-means clustering, which assesses the structure of the learned representations by clustering the feature space and evaluating the purity of the clusters with respect to known labels. While this does not impose a linear separability on the embedding space, if the representations are well-structured, features belonging to the same class should cluster together. Finally, a more heavy-handed approach is fine-tuning the model on a specific task and comparing the number of steps or epochs required to reach a particular performance level provides another means of evaluation. A representation that allows the model to quickly converge to high performance with minimal fine-tuning indicates that the initial self-supervised training has effectively captured the essential features of the data. By comparing these methods, researchers can determine how well the self-supervised model prepares the feature space for various tasks, with faster convergence in fine-tuning often highlighting a more versatile and robust representation. These evaluation techniques collectively offer a comprehensive assessment of the learned representations, guiding further refinement and optimisation.

2.5 Representation Learning in Computer vision

In this section, we will go in detail into the role and application of representation learning, specifically in the domain of computer vision.

2.5.1 The Diversity and Fragmentation of Computer Vision Research

Computer vision is a vast field of research and applications that involves a multitude of tasks that take visual signals as input. These tasks range from image classification, detection, and segmentation to depth estimation, surface normal estimation, colourisation, in-painting, super-resolution and many more. As a result of this diversity, historically the research in computer vision is highly fragmented in terms of its methods, interfaces, as well as pre- and post-processing steps.

While most computer vision tasks use the RGB pixels as input to mimic the human visual system, the output requirements for each tasks can vary widely. Broadly, the output format can be categorised into two categories: sparse and dense prediction. For instance, tasks like image classification or object detection only need to output a small number of bits of information to denote the presence and the location of objects included in the inputs. On the other hand, tasks like segmentation or

depth estimation require a dense prediction for every input pixel, while tasks such as in-painting, out-painting or super-resolution demand the outputs to be even larger than the inputs itself.

Furthermore, the pre- and post-processing steps in computer vision also vary widely. Pre-processing steps may include data augmentation and normalisation, while post-processing steps may involve smoothing or filtering heuristics such as non-maximal suppression [226] or conditional random fields [38]. The choice of pre- and post-processing steps can greatly affect the performance of the model on a particular task.

While the diversity of tasks provides ample opportunities for research and innovation, it also makes it difficult to develop general-purpose models that perform well across multiple tasks. In the section below we discusses some major research directions towards this goal.

2.5.2 Programming With Expert Models

Combining different models for various computer vision tasks is a straightforward approach to building a more complex visual agent. For instance, to classify human gaits, we can use a human keypoint detection model that takes an RGB input image and outputs a list of keypoints for each human in an image. We can then feed these keypoints to a gait classification model to generate the final gait prediction. This approach offers the advantage of allowing each task to be developed and researched independently, in parallel with the others.

While the independent of tasks allows them to be developed in parallel, this is also a disadvantage of this approach, because each model needs to be trained independently for each task. This is especially wasteful when tasks share the same inputs and are very similar in outputs. Take the task of depth estimation and object segmentation for example. Both tasks take an input image and predict either a depth value, or an object category for each input pixel. It is reasonable to assume that the information processing steps needed to predict depth are also useful to predict the object it belongs too, and vice versa. So training two independent networks to process the same information is inefficient, and does not allow the learning signal from one task to benefit the other and vice versa.

This approach to training separate expert models and chaining them together to perform complicated tasks is similar to the concept of Application Programming Interface (API) in system programming. Care must be taken to explicitly define the input and output interface of each model, but when properly organised, it is possible to compose and perform an exponential amount of tasks from a small set of expert models.

Recently, there have been proposals to leverage the natural language processing capabilities of large language models to program these computer vision model APIs [243]. By using a large language model acting as the controller to parse natural language descriptions, it can understand the input and output requirements of each expert model, and devise the appropriate sequence of inferences to obtain the desired result. This approach has the potential to greatly simplify the process of building complex computer vision agents by allowing developers to specify tasks in natural language instead of having to code them explicitly.

2.5.3 Perceiving With Visual Representation

The concept of a shared visual representation for a wide range of tasks has been a driving force behind recent progress in computer vision. The idea is inspired by the human's ability to perform multiple tasks from a common visual pathway without much specialisation.

One of the earliest attempts to leverage hidden representations for vision tasks was transfer learning [302]. Instead of training a deep network from scratch, researchers transferred the base of a network that had been trained on a similar task to extract the visual representation and then only trained task-specific components for the target task. Finetuning is a similar approach where the pretrained network is learned together with a new task. This requires more memory and computing power, but generally achieves better performance than transfer learning, especially when the target and source input distributions are different.

In the early days, the most popular backbones were image classification models trained on the ImageNet dataset [58]. Due to the size of ImageNet, the models trained on this supervised task were usually among the largest and most powerful models. Improvement in ImageNet pretraining consistently improved performance on a wide range of downstream tasks.

In recent years, self-supervised learning has taken over from supervised ImageNet pretraining, thanks to its lack of dependence on human labels [110]. This approach scales to even larger models trained on even bigger datasets, sometime comprising billions of images [92]. Due to their size and generic pretraining objectives, these backbones are sometime referred to as "foundation models" [23], to indicate their function and goal of supporting other models on downstream tasks.

Complementary to the effort of unifying the visual representation backbone is the research in unifying the pre- and post-processing and heuristics of various vision tasks. These approaches follow the direction of natural language processing, treating inputs and outputs as sequences of tokens and leveraging autoregressive models for next-token prediction to model the dependencies among them [154, 42]. Combined

with a foundation model for visual representation, this approach offers the potential for even greater flexibility and efficiency in multi-task learning.

2.5.4 Reasoning with Semantic Representation

The recent progress in computer vision research has been heavily reliant on large, pretrained foundation models that provide powerful visual representations. These models scale smoothly and reliably with more compute power and diverse data on the pretraining objective, and can sometimes lead to "emergent behaviours"[284] when evaluated on some metrics for downstream tasks. However, the cost of training these models is immense, taking weeks or even months on some of the world's most powerful supercomputers and requiring datasets larger than any human could see in their lifetime. This begs the question: is scaling all you need ? Even if it is, how much more is needed to achieve the ultimate goal of creating a general purpose autonomous agent that can perceive, navigate, and interact effectively in the real world ?

While perception tasks, which involve generating labels or masks, can be handled by pretrained visual representation backbones, there are other types of tasks that require more than just perception. These include tasks like understanding and reasoning about the objects in an image, as well as reasoning about counterfactual scenarios. How can we approach building such systems that can reason and interact with the world like human do ?

The concept of "System 1" and "System 2" thinking, as introduced by Daniel Kahneman in his book "Thinking, Fast and Slow" [139], offers a useful framework for understanding the different types of cognitive processing that occur in the human brain. System 1 thinking is fast, automatic, and often subconscious, while System 2 thinking is slow, deliberate, and conscious. Drawing on this analogy of human psychology, the current limitations of visual representation are similar to the limitations of System 1. The development of higher-level representations and the ability to perform System 2 processing in deep learning models is a promising direction for advancing the field towards more advanced cognitive capabilities.

One approach to achieving more advanced capability is to develop higher-level representations that can capture abstract concepts such as objects automatically. These object-centric representation can be thought of as analogous to the working memory in humans, which can hold only a few distinct concepts at a time [50].

In the context of programming, this can be likened to the way simple computations are assigned to variables, which can then be manipulated to compute more complex operations. By forming these higher-level representations, models can bind concepts to variables and perform reasoning on them.

The benefits of such higher-level systems are numerous. Instead of learning to represent a complicated scene in a single vector representation, which would require an enormous amount of data to cover the exponential number of configurations, such a system can learn to discover and represent simpler "object" constituents. The model can later learn to compose them to represent a combinatorial amount of scenes, making it more efficient to learn in terms of compute and data. In essence, such a system can provide the model with the ability to break down complex scenes into smaller, more manageable parts, making it easier to learn and reason about the visual world.

Moreover, being able to discover and learn such representations provides a way to bridges the gap to symbolic reasoning, which involves using symbols as vectors and reasoning as arithmetic. This ability enables end-to-end optimisation of perceiving and reasoning, which is a critical aspect of building an autonomous agent that can navigate and interact with the real world effectively. By learning to reason symbolically, the model can make more complex inferences and predictions, and it can do so in a more efficient and effective manner. This capability is crucial for building agents that can understand and reason about the world in a way that is comparable to human cognition.

This is an exciting research direction with many open questions and challenges ahead.

2.6 Towards a Generally Intelligent Agent

In this chapter, we have presented a brief overview the rich landscape and foundational concepts in the domains of computer vision and machine learning, with a particular emphasis on deep neural networks. These are all integral components of the broader field of Artificial Intelligence. We have delved into the historical evolution of computer vision, tracing how it has matured to enable machines to perceive and comprehend the visual world. Furthermore, we have tried to unravel the effectiveness and ubiquity of artificial neural networks within the context of machine learning and deep learning, highlighting its pivotal role in emulating human-like capabilities in some cognition tasks, and the various architectural paradigms that have emerged as a result, from Multi-layer Perceptrons, Convolutional Neural Networks to Transformer models. These advances have transcended mere theoretical constructs, manifesting as a myriad of practical and impactful applications across diverse domains and exerting a profound influence on the technology industry as a whole.

In Section 2.4, we have spotlighted the critical theme of representation learning, a cornerstone principle underpinning the remarkable performance gains achieved by

deep learning across a spectrum of tasks. After all, the true measure of intelligent behaviour lies in its ability to operate effectively within a given environment, and perceiving and representing that environment constitutes a foundational step towards achieving this objective. Representation learning serves as the principle for machines to encode and comprehend information, mirroring a fundamental aspect of human cognition. We've explored how these representations form the basis for acquiring knowledge, enabling models to learn more general representations that can adapt to a wide range of tasks and domains. The pursuit of different approaches to representation learning, essential for comprehending and enhancing this pivotal concept, lies at the heart of all the research questions posed in this thesis, as outlined in Section 1.1.

This section serves as the cornerstone for the remaining chapters of this thesis, upon which the subsequent methodologies for self-supervised learning, contrastive representation learning (Chapter 3), and object-centric representation (Chapter 4) are built.

Zooming in on the realm of computer vision, we have delved into an emerging paradigm for developing more capable vision systems and the possible role of representation learning, as detailed in Section 2.5. Departing from the practice of constructing individual representation spaces for specific tasks, the approach of pre-training a general visual representation on vast volumes of data, alluded to in Section 2.5.3, is the focus of study in Chapter 3.

Furthermore, the impetus to encourage models to acquire highly compact and abstract representations represents a promising trend in addressing the challenge of high-level visual reasoning, as introduced in Section 2.5.4.

In the grander scheme of things, the ultimate objective of both computer vision and machine learning is to advance towards a more generally intelligent agent. Irrespective of the internal representations learned, what holds most significance is an agent's capacity to make accurate predictions and decisions based on incoming signals. While the dominant *modus operandi* in the era of deep learning has been more layers, more computational resources, and larger datasets, an examination of the problem through the lens of the principles underpinning representation learning may yield invaluable insights and hasten progress towards this overarching goal.

Chapter 3

Contrastive Representation Learning: A Framework and Review

In this chapter, we address **research question 1** regarding how to learn a general visual representation. We provide a comprehensive review and analysis on the topic of Contrastive Representation Learning (CRL), a self-supervised discriminative method that can learn visual representations that outperform supervised pre-training on a wide range of downstream tasks.

Through the lens of contrastive learning, we address **research question 2** by categorising the many different learning principles and inductive biases that enable learning useful visual representations. While this chapter focuses on contrastive learning, the principles and framework provided are generally applicable and can be applied to a broader spectrum of topics in visual representation learning.

In the following sections, we first introduce the concept of CRL via a concrete example of Instance Discrimination in Section 3.1.1. We then introduce the general framework of CRL and the taxonomies of its components in Section 3.2. Using this framework, we then provide the pertinent historical evolution of the development of contrastive learning and its application in a wide range of domains and modalities. We then conclude the chapter with a discussion on future research directions, as well as pointers for practitioners looking to apply CRL.

3.1 What is Contrastive Representation Learning ?

We now present an intuitive introduction to contrastive learning with a concrete example of the *Instance Discrimination* task in learning self-supervised visual rep-

resentations.

Intuitively, contrastive representation learning can be considered as learning by comparing. Unlike a discriminative model that learns a mapping to some (pseudo-)labels and a generative model that reconstructs input samples, in contrastive learning a representation is learned by comparing among the input samples. Instead of learning a signal from individual data samples one at a time, contrastive learning *learns by comparing* among different samples. The comparison can be performed between positive pairs of "similar" inputs and negative pairs of "dissimilar" inputs.

Unlike supervised methods where a human annotation y is needed for every input sample $\mathbf{x}$, contrastive learning approaches only need to define the rules, or distribution, of similarity in order to sample a positive input $\mathbf{x}^+ \sim p^+(\cdot|\mathbf{x})$, and a rule or data distribution for a negative input $\mathbf{x}^- \sim p^-(\cdot|\mathbf{x})$, with respect to an input sample $\mathbf{x}$. The goal of contrastive learning is very simple: the representation of "similar" samples should be mapped close together, while that of "dissimilar" samples should be further away in the embedding space. Thus by contrasting between samples of positive and samples of negative pairs, representations of positive pairs will be pulled together while representations of negative pairs are pushed far apart.

In the self-supervised setting, instead of deriving a pseudo-label from the pretext task, contrastive learning methods learn a discriminative model on multiple input pairs, according to some notion of similarity. Similar to other self-supervised pretext tasks, this definition of similarity can be defined from the data itself, and thus can overcome a limitation encountered in supervised learning settings where only a finite number of label pairs are available from the data. While some self-supervised methods need to modify the model architecture during learning (such as in [310]), contrastive methods are much simpler where no modification to the model architecture is needed between training and fine-tuning to other tasks.

If additional labels are provided, these can also be integrated into the definition of similarity and dissimilarity of the contrastive framework as well. By defining the similarity and dissimilarity distribution on the dataset level instead of individual data samples, contrastive methods alleviate the need for a labelled dataset while providing a mechanism to specify the desired invariant / covariant properties of the learned mapping. Thus contrastive learning methods provide a simple yet powerful approach to learning representations in a discriminative manner in both supervised or self-supervised setups.

Figure 3.1 illustrates the family of contrastive methods along generative-discriminative and supervised-unsupervised axes.

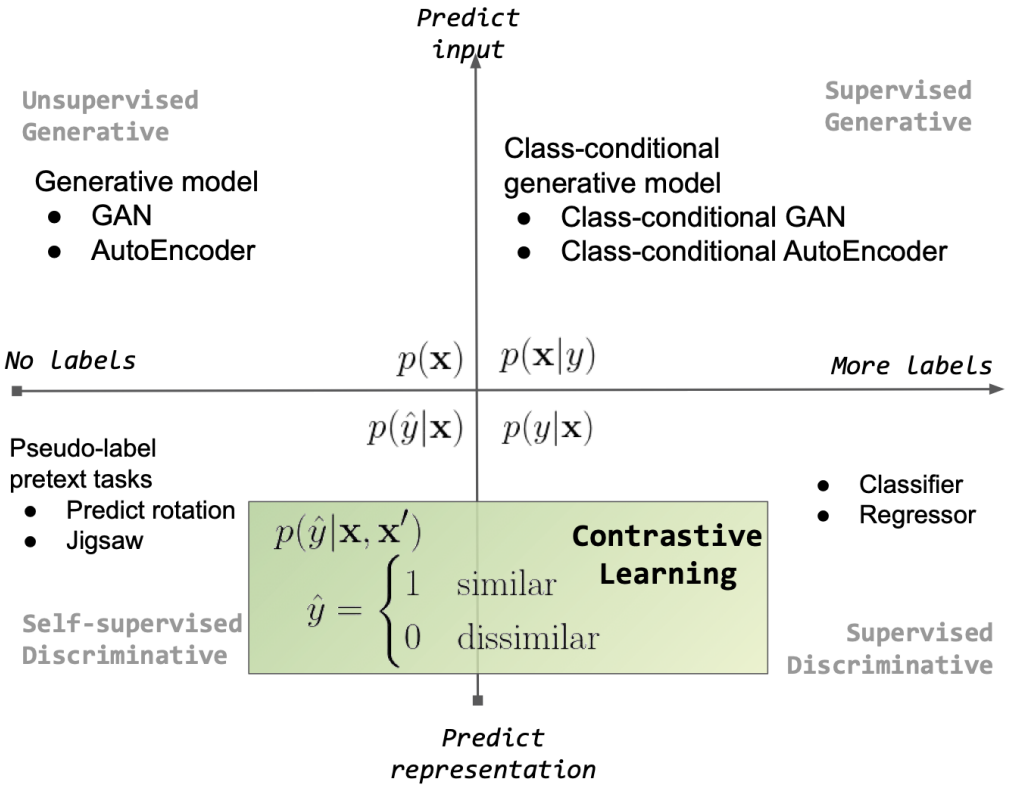

Figure 3.1: Contrastive learning in the Generative-Discriminative and Supervised-Unsupervised spectrum. Contrastive methods belong to the group of discriminative models that predict a pseudo-label of *similarity* or *dissimilarity* given a pair of inputs.

3.1.1 Example: Instance Discrimination

Along the lines of an exemplar-based classification task [68], which treats each image as its own class, Instance Discrimination [289] is a popular self-supervised method to learn a visual representation and has succeeded in learning useful representations that achieve state-of-the-art results in transfer learning for some downstream computer vision tasks [110], [190]. Based on the simple formulation proposed in SimCLR [41], in this section we will describe the Instance Discrimination task as a simple form of contrastive learning, as illustrated in Figure 3.2.

The image-based instance discrimination pretext task learns a representation by maximising agreement of the encoded features (embeddings) between two differently augmented views of the same images, while simultaneously minimising the agreement between views generated from different images. To avoid the model maximising agreement through low-level visual cues, views from the same images are generated through a series of strong image augmentation methods.

- Let $\mathcal{T}$ be the set of *image transformation* operations where $t, t' \sim \mathcal{T}$ are two different transformation operators independently sampled from $\mathcal{T}$. These

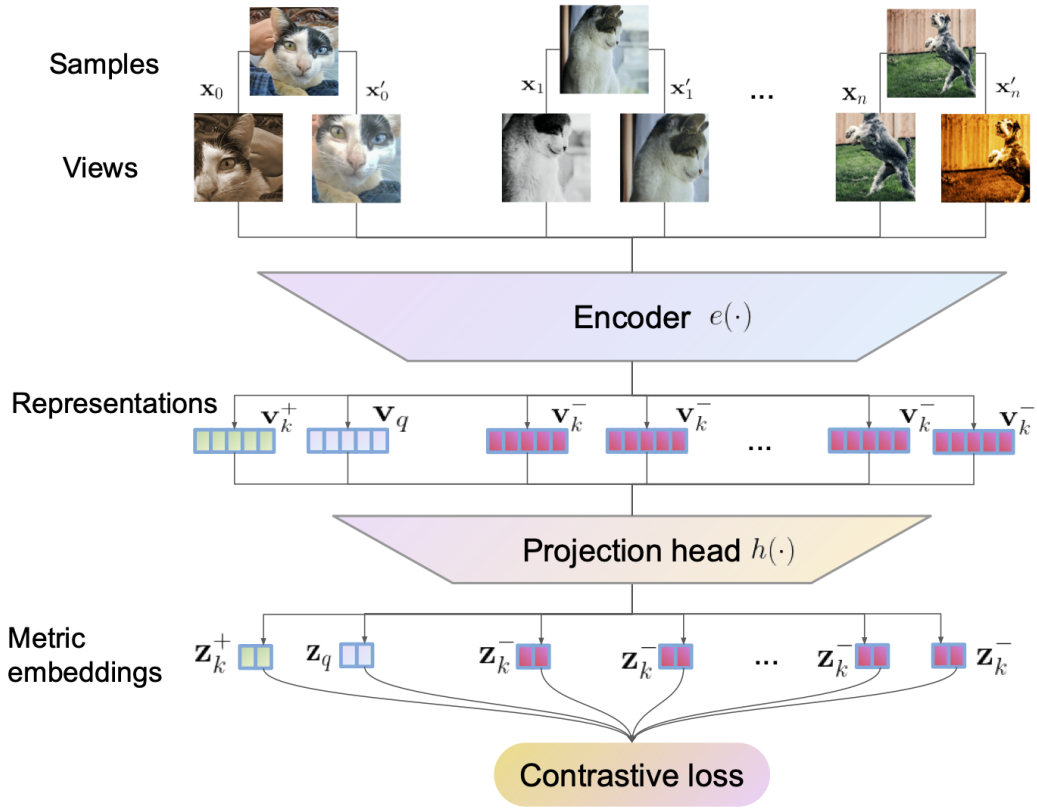

Figure 3.2: Contrastive learning in the Instance Discrimination pretext task for self-supervised visual representation learning. A positive pair is created from two randomly augmented views of the same image, while negative pairs are created from views of two different images. All views are encoded by a the same encoder and projection heads before the representations are evaluated by the contrastive loss function.

transformations could be random *cropping* and *resizing*, *blur*, *colour distortion* or *perspective distortion*, etc. A $(\mathbf{x}_q, \mathbf{x}_k)$ pair of query and key views is positive when these two views are created by applying different transformations on the same image $\mathbf{x}$: $\mathbf{x} = t(\mathbf{x})$ and $\mathbf{x}' = t'(\mathbf{x})$, and is negative otherwise.

- A *feature encoder* $e(\cdot)$ then extracts the feature vectors from all the augmented data samples $\mathbf{v} = e(\mathbf{x})$. There is no restriction on the choice of the encoder, so usually a simple CNN such as ResNet [108] or ViT [67] is used for image data due to their favourable performance characteristics. The representation $\mathbf{v} \in \mathbb{R}^d$ in this case is the output of the average pooling layer of Resnet.

- Each representation v is then fed into a *projection head* $h(\cdot)$ comprised of a small multi-layer perceptron (MLP) to obtain a metric embedding $\mathbf{z} = h(\mathbf{v})$, where $\mathbf{z} \in \mathbb{R}^{d'}$ with $d' < d$ is in a lower dimensional space than the representation $\mathbf{v}$. This projection head can be as simple as a one-layer MLP using a non-linear activation function. All the vectors are then normalised to be unit vectors.

- A batch of these metric embedding pairs $\{(\mathbf{z}_i, \mathbf{z}'_i)\}$, with $(\mathbf{z}_i, \mathbf{z}'_i)$ represents the metric embeddings from two augmented versions $(\mathbf{x}, \mathbf{x}')$ of the same image, are then fed into the *contrastive loss* function which encourages the distance in the metric embedding of the same pair to be small, and the distances of embeddings from different pairs to be large. The non-parametric classification loss [289] and its variants, such as InfoNCE [204] and NT-Xent [41] is a popular choice for the contrastive loss function, which for the i-th pair has the general form:

$$\mathcal{L}_i = -\log \frac{\exp(\mathbf{z}_i^\top \cdot \mathbf{z}'_i/\tau)}{\sum_{j=0}^{K} \exp(\mathbf{z}_i \cdot \mathbf{z}'_j)/\tau)} \tag{3.1}$$

  where $\mathbf{z}^\top \cdot \mathbf{z}'$ is the dot product between two vectors and τ is a temperature hyper-parameter that controls the sensitivity of the product. The sum in the denominator is computed over one positive and K negative pairs in the same minibatch. Intuitively, this can be understood as a non-parametric version of $(K+1)$-way softmax classification [289] of z_i to the corresponding z'_i.

In order to minimise the InfoNCE loss function in Eq. (3.1), the dot product in the numerator measuring the similarity of representation from the same pair is maximised, while the similarity of all negative pairs in the denominator, is minimised.

When the contrastive training phase is done, the projection head is discarded and the encoder is used as the feature extractor for transfer learning. By combining the predictor or classifier with the representation output of the encoder, they can be fine-tuned on a new task on a target dataset.

Contrastive methods in the instance discrimination task set out to learn a representation that can separate between different instances, while ignoring the meaningless variances introduced by image data augmentation. Because contrastive learning directly maximises similarity between representations of positive similar pairs and minimises that of negative pairs, how those pairs are generated directly determines the invariant properties in the learned representation. The most important component for the success of contrastive pre-training on ImageNet [58] is data augmentation methods. As analysed in SimCLR [41], many contrastive methods perform very poorly without proper augmentations (i.e random crop and colour distortion) even for the same set of architectures and losses.

The dataset, data transformations and instance-wise similarity definition combined together in the contrastive learning framework provide a scalable and accessible approach to specifying invariant and covariant properties in the learned representation.

3.2 A Taxonomy for Contrastive Learning

Before we present our taxonomy for contrastive learning methods, we first formally describe the contrastive representation learning (CRL) framework in Section 3.2.1. In particular, the CRL is a general framework that can be used to succinctly describe a variety of contrastive learning methods ranging from self-supervised to supervised and covering images, videos, audio, text and more. We use this framework to introduce a comprehensive taxonomy for the components of contrastive methods in Sections 3.2.2, 3.2.3, 3.2.4 and 3.2.5.

3.2.1 The Contrastive Representation Learning Framework

The general CRL framework, illustrated in Figure 3.3 builds on top of SimCLR of Chen *et al.* [41], which describes a simple contrastive self-supervised framework to learn visual representations in the context of an Instance Discrimination task (see Section 3.1.1). As distinct from SimCLR, we generalise this framework beyond the image Instance Discrimination task to cover learning representations in a variety of data domains (images, video, audio and text), learning setups (supervised, self-supervised or knowledge distillation) and ways to define the concept of similarity. Specific choices of the similarity distribution, encoders and heads as well as contrastive loss functions allows the CRL framework to encompass arbitrary contrastive learning methods. More importantly, it enables a clear understanding of most of the contemporary work and sheds light on the limitations and the promising directions ahead.

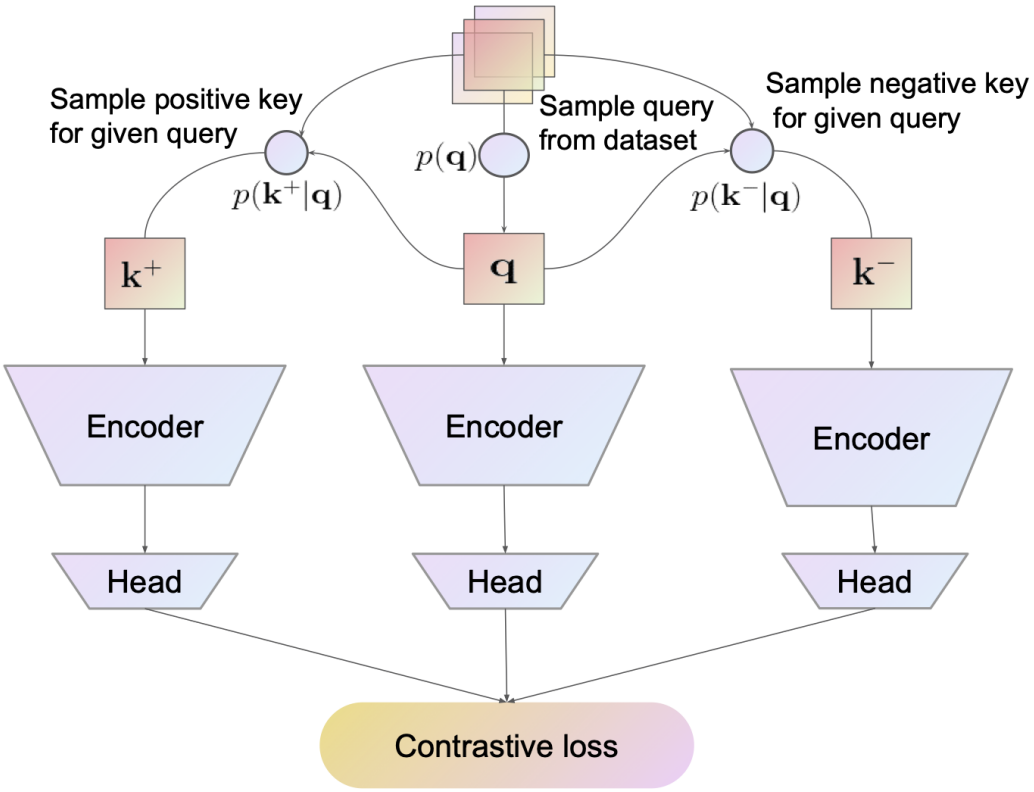

Figure 3.3: Overview of the Contrastive Representation Learning framework. Its components are: a similarity and dissimilarity distribution to sample positive and negative keys for a query, one or more encoders and transform heads for each data modality and a contrastive loss function evaluate a batch of positive and negative pairs.

In the following and throughout the rest of this thesis, we adopt the metaphor of *query* and *key* from He et al. [110], inspired by the problem of similarity matching as a form of dictionary look-up similar to terminology used to describe the attention layer.

We will use the symbols q and k to represent the *query* and *key* for either the input sample $\mathbf{x}$, the representation $\mathbf{v}$ or the metric embedding $\mathbf{z}$ depending on context. When we need to be specific, the corresponding symbols $\mathbf{x}, \mathbf{v}, \mathbf{z}$ with superscript $\cdot^q, \cdot^k$ for query and key will be used.

Definition 3.2.1 (Query, Key)**.** *Query* and *key* refer to a particular view of an input sample $\mathbf{x} \in \mathcal{X}$. Together they form a positive or negative pair $(\mathbf{q}, \mathbf{k})$ depending on whether the query and key are considered similar or not.

In the Instance Discrimination task, query and key views are a randomly transformed version of an image $t(\mathbf{x})$ in the data set $\mathcal{X}$.

Definition 3.2.2 (Similarity distribution)**.** A *similarity distribution* $p^+(\mathbf{q}, \mathbf{k}^+)$ is a joint distribution over a pair of input samples that formalises the notion of similarity (and dissimilarity) in the contrastive learning task. Distinct from other machine learning methods where the data distribution is defined over a single input sample $p(\mathbf{x})$, the *similarity* required by contrastive methods takes input from the joint distributions of pairs of samples $p(\mathbf{q}, \mathbf{k})$.

A key is considered positive $\mathbf{k}^+$ for a query $\mathbf{q}$ if it is sampled from this similarity distribution and is considered negative k^- if it is sampled from the dissimilarity distribution $p^-(q, k^-)$. In some tasks, the dissimilar data distribution may not be explicitly defined but implicitly given as the distribution of any pair that is not sampled from the similarity distribution.

Similar to other representation learning problems, the focus of contrastive learning is in learning from a high-dimensional input space $\mathcal{X}$, which depends on the domain and can be a tensor representing audios, images, videos or texts.

Combining the data distribution $p(\mathbf{x})$, the definition of similarity $p^+(\mathbf{q}, \mathbf{k})$ and dissimilarity $p^-(\mathbf{q}, \mathbf{k}^-)$, different properties of the learned representation can be specified, as illustrated in Figure 3.4.

In practice, queries and keys are not necessarily sampled jointly but the query can be sampled first from the data distribution $\mathbf{q} \sim p(\mathbf{x})$ where the corresponding positive and negative keys are then sampled from the conditional distributions $\mathbf{k}^+ \sim p^+(\cdot|\mathbf{q})$ and $\mathbf{k}^- \sim p^-(\cdot|\mathbf{q})$.

In the Instance Discrimination task, the similarity distribution is defined over any pair that are transformed from the same input samples $\mathbf{q}, \mathbf{k} \sim p^+(\cdot, \cdot)$ if $\mathbf{q} = t(\mathbf{x})$ and $\mathbf{k} = t'(\mathbf{x})$ for 2 different random transformations t and $t' \in \mathcal{T}$.

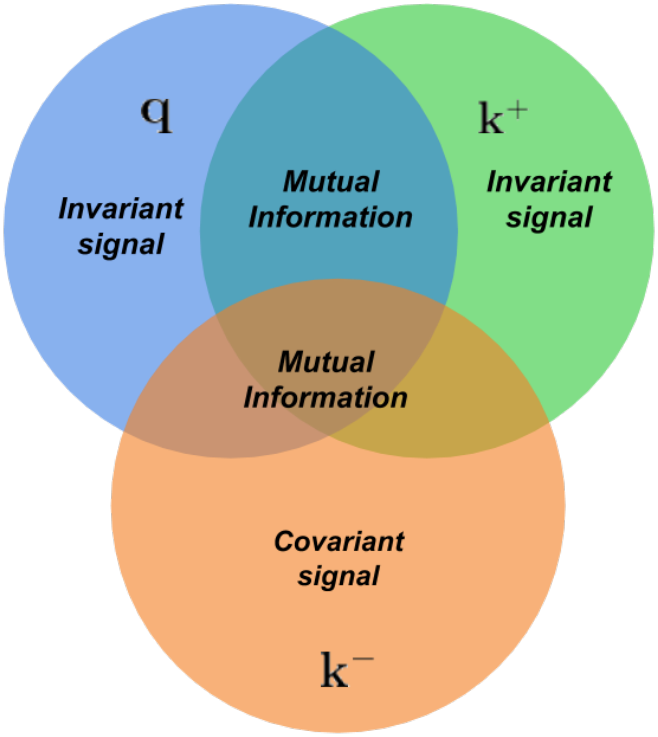

Figure 3.4: An intuitive diagram represents the learning signal captured by the contrastive loss through the query, positive and negative keys. Contrastive methods allow the desired invariances to be specified through the similarity and dissimilarity distributions. Each circle represents the information signal contained in each view. The signal that is not mutual between query and positive keys are invariant features, since their representations are made as similar as possible. The signal that is not mutual between the negative key and the query or positive keys are covariant features, since these representations must be able to distinguish between those to minimise similarity to the negative key.

Definition 3.2.3 (Model)**.** We refer to the combination of all modules with parameters in a contrastive learning method as the **model** $f(x;\theta) : \mathcal{X} \to \mathbb{R}^{|\mathcal{Z}|}$ and its parameters collectively as θ.

The model can be decomposed further into a base encoder and a transform head.

Definition 3.2.4 (Encoder)**.** The features *encoder* $e(\mathbf{x};\theta_e) : \mathcal{X} \to \mathcal{V}$ with parameters θ_e learns a mapping from the input views $x \in \mathcal{X}$ to a representation vector $v \in \mathbb{R}^d$. This network (when trained via contrastive learning) can be used to generate features (or inputs) to leverage the learned representations in other tasks (e.g. as input when learning another model for an image classification task), or to have layers stacked on top (e.g. fully connected, softmax) where the network can be fine-tuned to the new task.

Definition 3.2.5 (Transform head)**.** *Transform heads* $h(\mathbf{v};\theta_h) : \mathcal{V} \to \mathcal{Z}$ parameterised by θ_h, are modules that transform the feature embedding $\mathbf{v} \in \mathcal{V}$ into a metric embedding $\mathbf{z} \in \mathbb{R}^{d'}$.

Depending on the specific application, the transform heads can be used for different purposes, such as to aggregate information from multiple representation vectors or to project it down to a lower-dimensional space the contrastive loss.

Definition 3.2.6 (Contrastive loss)**.** A contrastive loss function operates on a set of metric embedding pairs $\{(z, z^+), (z, z^-)\}$ of the query, positive and negative keys. It measures the similarity (or distance) between the embeddings and enforces constraints such that the similarity of positive pairs are high and the similarity of negative pairs are low. To attain small distances between the embeddings of positive pairs in the metric space, representations will become **invariant** to irrelevant differences in the input space of positive pairs, while simultaneously learning the **covariant** representation between negative pairs to explain for the large distance in the metric space.

3.2.2 A Taxonomy of Similarity

Contrastive Learning revolves around learning a mapping from different views of the same *scene*, or *context* into the same region of a representation space, which is formalised through a similarity distribution. The key to an effective contrastive learning task is to design the similarity distribution such that positive pairs are very different in the input space yet semantically related, and a dissimilarity distribution such that negative pairs are similar in the input space but semantically different. Despite the recent popularity of self-supervised contrastive learning, contrastive learning in general is agnostic to the supervised / unsupervised paradigm.

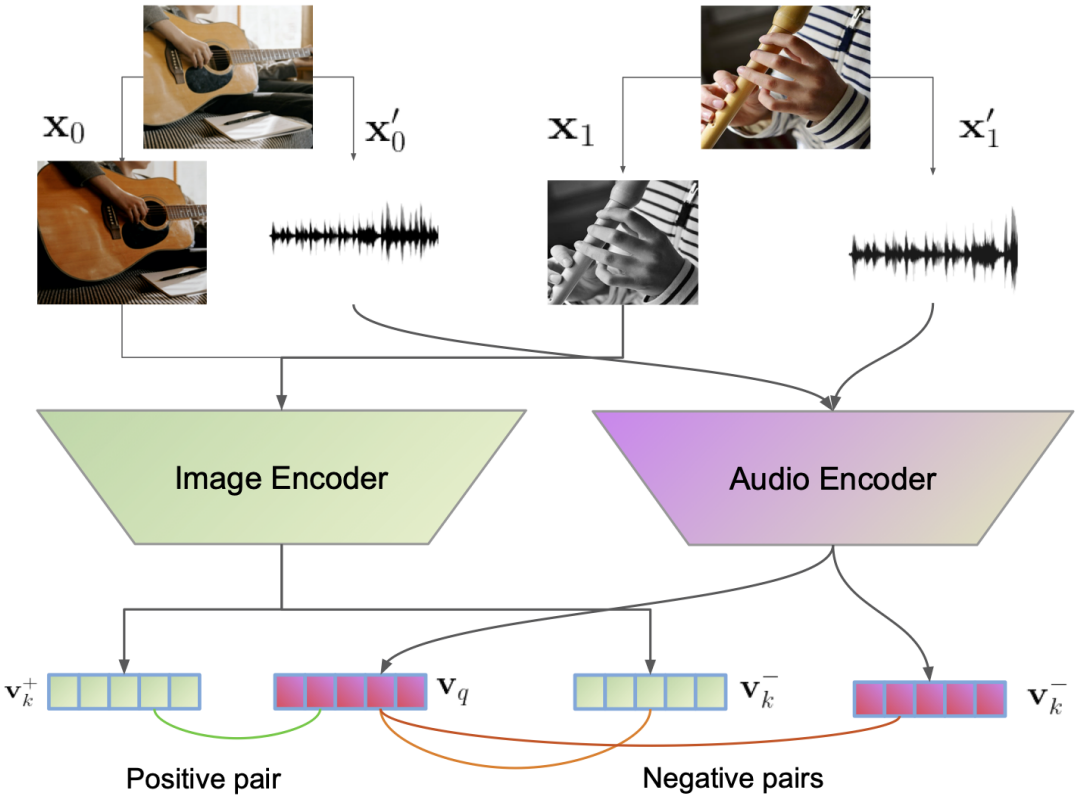

Figure 3.5: Illustration of learning similarity between multiple modalities. Each modality has an encoder and the representations extracted by different encoders are contrasted with each other to learn a joint embedding space.

Depending on whether any human labels y are used in defining those joint distribution, e.g. $k \sim p(\cdot|q, y)$, the method then becomes a supervised or self-supervised contrastive learning task.

Depending on the end goals there can be many notions of similarity and dissimilarity, which is a strong point of contrastive methods, but it also makes it difficult to provide a taxonomy that captures all these variations. However, there are some general principles that are usually the underlying assumptions behind how similarity and dissimilarity is constructed, which we now examine.

Multisensory signals

One direct approach to have multiple views of the same context is to record the information with multiple sensors. These sensors can be of the same modality (e.g. two cameras recording the same scene from different angles), or of different modalities (e.g. audio and image from a video), as illustrated in Figure 3.5. Using the natural correspondence between different sensors, the model can learn to be invariant to the low-level details in each sensor input and focus on representing the shared context between them.

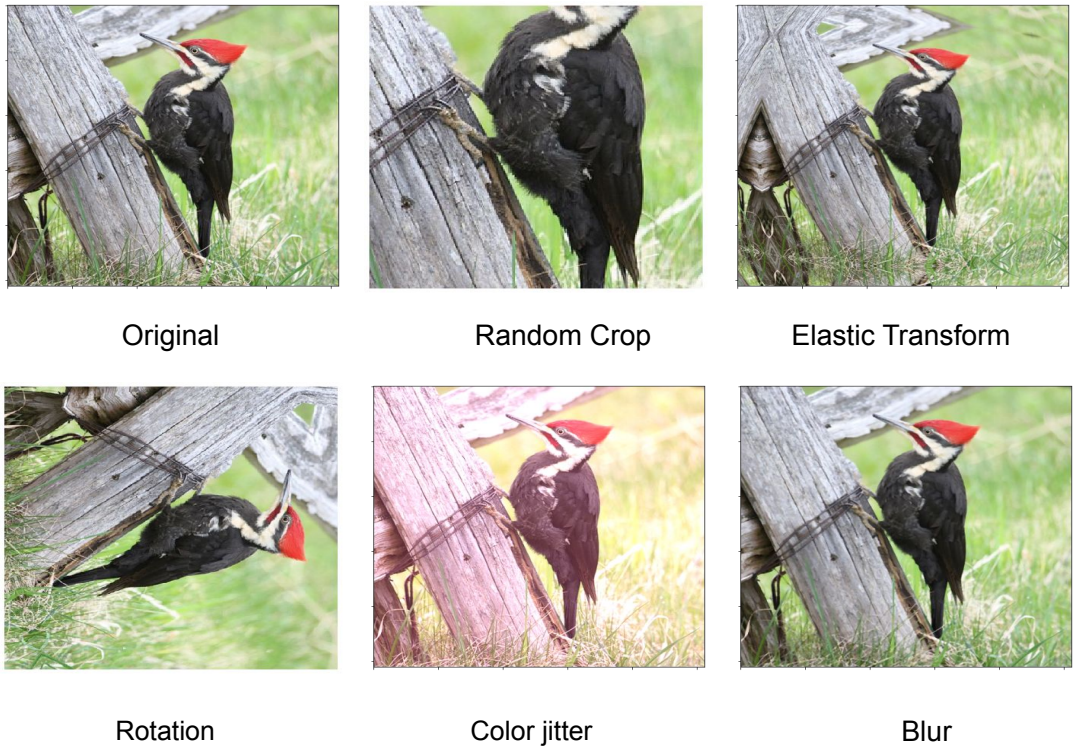

Figure 3.6: Illustration of some common image augmentation methods. Different views from a random set of augmentations of the same images are usually considered positive pairs.

Contrastive methods have been used to learn cross-modal representations of visual and textual data in [261], [126]. In the Time-Contrastive Network [242], a visual representation is learned by pulling the representation of two simultaneous views from *different cameras* of the same scene, while pushing apart *frames taken from far away in time but from the same video*. This leads to a representation space that is invariant to viewpoints while being sensitive to changes in time.

Data transformation

If synchronous data from multiple sensors is not available (e.g. a single-modality dataset like ImageNet), the most simple yet effective approach to generating different views of the same scene is to use a hand-crafted transformation function operating on the input data domain. Designing and implementing such semantic-preserving transformations requires prior knowledge, but this knowledge is defined once for the entire dataset or data collection pipeline, and can be dynamically applied to individual samples at run time.

For visual data, image augmentation methods such as lighting or colour distortion, cropping and padding, adding noise and blur, rotation and perspective

transformations, etc. are efficient methods to transform pixels while preserving the semantic meaning of an image's content. An example of these data transformations techniques on image can be seen in Figure 3.6. Destroying low-level visual cues by image augmentation forces the contrastive method to learn a representation invariant to those changes in the inputs. These techniques have been widely used in supervised learning to learn invariant features and to increase the robustness of the resulting models. The recent wave of instance discrimination contrastive methods have demonstrated that the same representation can be learned from these augmentation techniques without the need for a class label [289, 300, 190, 110, 41].

For natural language text data, Fang et al. [77] transform a sentence using a back-translation method to create a slightly different sentence that has the same semantic meaning as the original one to form a positive pair. Back-translation uses two machine translation models to translate a sentence into a target language and back to the source language. The randomness from the two translation models will yield a sentence in the source language that is slightly different from the original sentence.

For program code data, ContraCode [128] uses various source-to-source transformation methods from the compiler literature such as variable renaming, identifier mangling, reformatting, beautification, compression, dead-code insertion / elimination, etc. to construct semantically similar code snippets that share the same functionality. Learning to map these textually different but functionally equivalent programs to the same feature vector allows the model to learn a function representation space that is predictive of equivalent programs.

For audio data, some augmentation methods such as *warping*, *frequency and temporal masking* in the Mel spectrogram format could be used to create different version of the same audio data, as in [197].

Context-instance relationships

Another approach to extracting similar views of the same scene is by exploiting the context-instance relationship from a sample representation. Generally, we want to learn a representation that captures the entire context, i.e the global information about a scene. That context can usually be decomposed further into parts, each containing a subset of the scene's information that is local to each subset.

Explicitly constraining the representation of the parts (local features) to be similar to the representation of the whole (global features), while being different from the representation of other views is a clever approach to defining similarity. Contrasting between the representation of local features versus global features can encourage the model to learn important features that present in the local views, while ignoring noise features which occur only in those local inputs. Representation from local

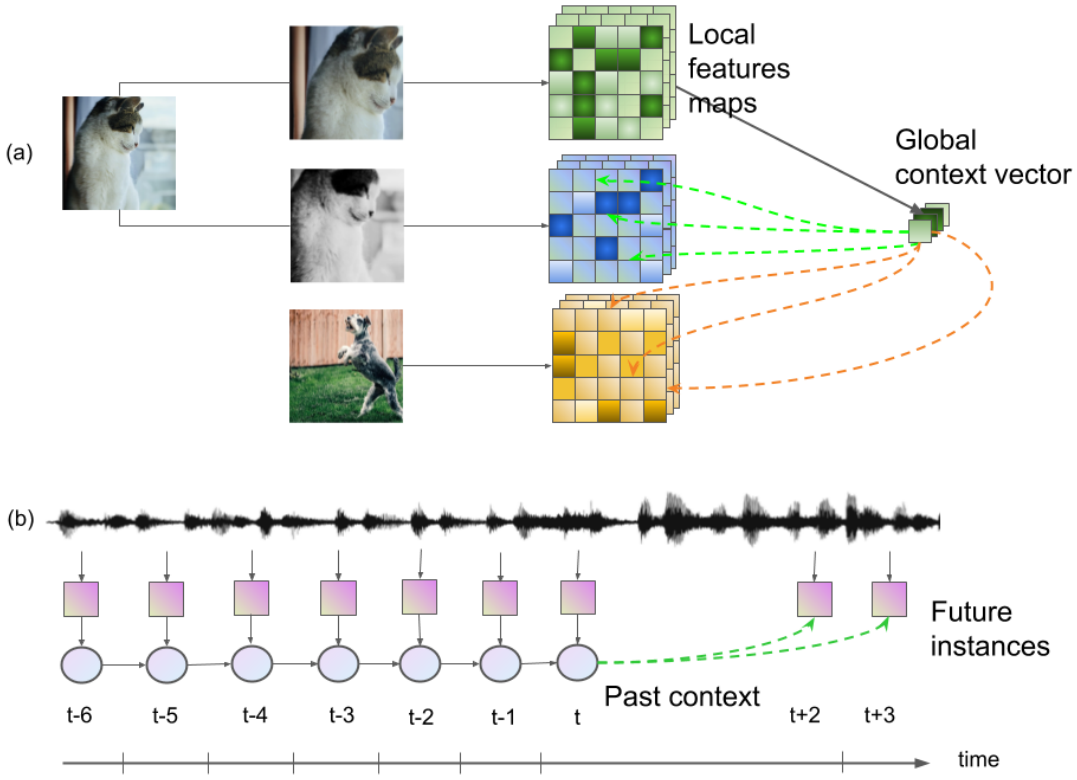

Figure 3.7: Illustration of extracting query and keys using the context-instance relationship. In *a)*, the context is a global summary vector of the entire image, while the instances are the local features in the set of intermediate feature maps. In *b)*, the past context is aggregated with a RNN contextualisation head and the instance are representations of future time steps.

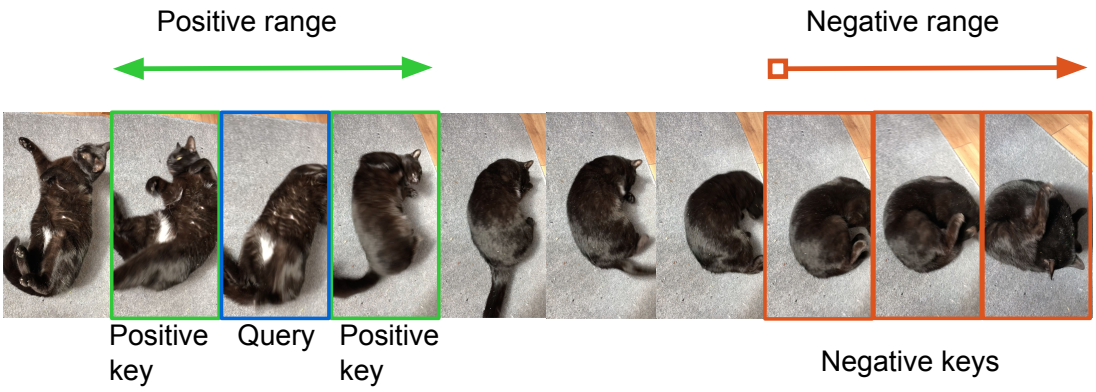

Figure 3.8: Illustration of sampling query and keys using the sequential coherence property of video data. The positive keys are defined as frames inside a small window surrounding the query frame. The negative keys are frames from the same video but are far away in time to the query.

features is thus encouraged to capture meaningful information relevant to the whole context, while global features are encouraged to capture as much detail from the local instances as possible.

Figure 3.7a describes the approach taken in Deep InfoMax (DIM) [118], where an image is encoded into a global feature vector and also into a feature map corresponding to spatial patches of pixels in the original image. The global feature and local features in the feature map of the same images then form positive pairs, while global features with local features from other images are considered negative pairs.

Global features can also be constructed from videos in the temporal dimension, as in Figure 3.7b. In Contrastive Predictive Coding (CPC) [204], context features are constructed as a summary of past input segments, and then contrasted with local features from a future time step. Contrastive learning to predict the correct future from the past context in this way can be thought of as an instantiation of the predictive coding theory.

Sequential coherence and consistency

In addition to the context-instance feature relationship, exploiting the spatial or temporal coherence and consistency in a sequence of observations is another approach to defining similarity in contrastive learning. This method works for a data domain that can be decomposed into a sequence of smaller units, such as an image into a sequence of pixels, or a video into a sequence of frames, etc. The representation of continuous views in a sequence is considered as a positive pair while discontinuous and far away pairs in the same sequence or different sequences are considered negative pairs. This approach uses the *slowness assumptions* in representation learning, which states that important features are the ones that change slowly over a sequence

of observations. Therefore, by learning invariant, slowly changing features in a sequence, a model will learn to extract the most important features in the data, as illustrated in Figure 3.8.

Rather than using simultaneous videos with multiple viewpoints as in Time-Contrastive Networks (TCN) [242], [70] uses a *multi-frame TCN* that exploits the temporal coherence property of video and applies contrastive learning on a sequence of frames, where frames inside a time-window are positive to each other, and pairs from with a frame outside the window are considered negative.

In addition to the hand-crafted transformations described in Section 3.2.2, the temporal coherence of video frames can also provide a natural source of data transformations. In a video, an object can undergo a series of transformations such as object deformation, occlusion, changes in viewpoint and lighting. These methods have been used in [277, 216] to learn representations of objects from videos without any additional labels.

Natural clustering

Clustering is the process of finding high-level semantics for groups of instances features according to some distance measure in the embedding space. Natural clustering refers to the assumption that different objects are naturally associated with different categorical variables, where each category occupies a separate manifold in a representation space. The distance between different clusters loosely represents the similarity between categories. This assumption is consistent with how humans naturally categorise and name different groups of objects, and is an important assumption in unsupervised learning, manifesting itself in various clustering algorithms such as K-Nearest neighbours. Semantic class labels in classification problems are also an instance of this assumption where the number of clusters and the names for these clusters are given by human annotators. Each cluster represents a high-level semantic concept and together the set of clusters provide overall structure to the data manifold.

Contrastive learning induces a metric in the embedding space where positive pairs have smaller distances between them and negative pairs have large distances, based on a semantic definition of similarity. In contrast to clustering methods which enforce the cluster assumption in a top-down fashion, contrastive methods enforce local smoothness between positive pairs thus organising the embedding manifold from the bottom up. Since contrastive learning and clustering methods essentially encode the same assumption but from different directions, the combination of contrastive methods from bottom-up and clustering approaches from top-down are a promising approach which complement each other's advantages. Figure 3.9 demonstrates this idea of combining contrastive learning with clustering methods.

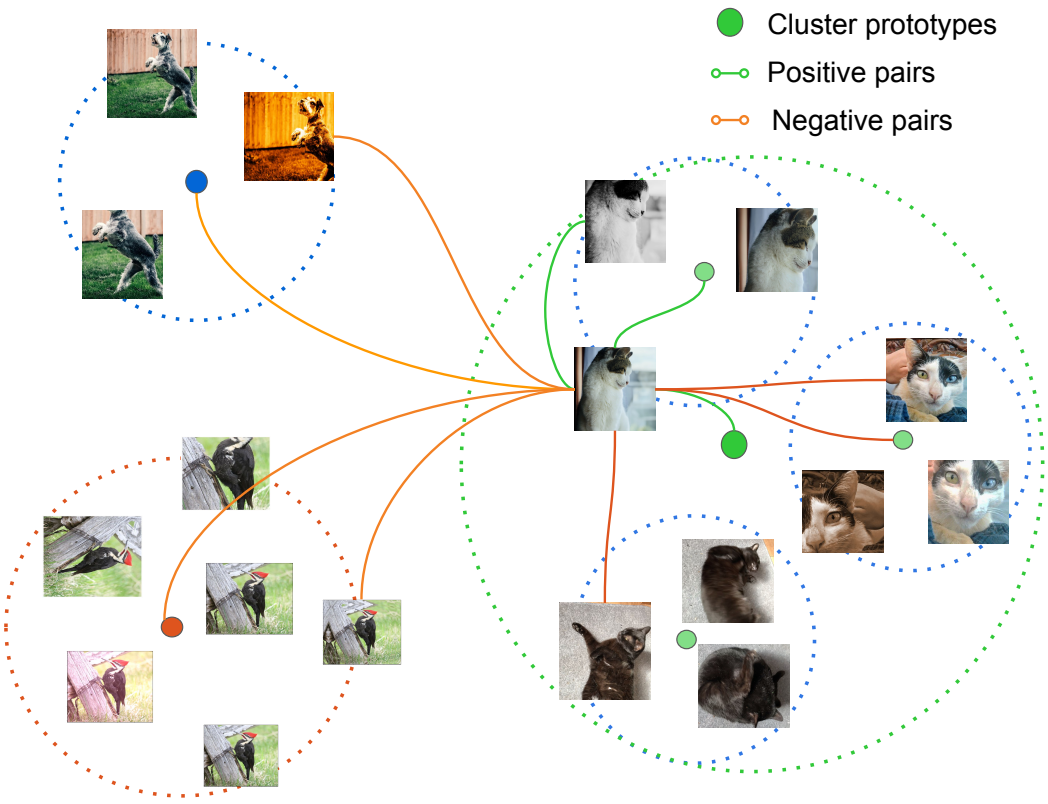

Figure 3.9: Illustration of contrastive methods on clusters. In addition to an individual sample's vector, there can also be cluster prototypes with different levels of granularity. Contrastive loss can operate on both the sample and cluster level.

Many different methods have tried to use contrastive methods to learn invariant properties while supplementing higher-level semantic information to the contrastive framework using clustering methods, such as Prototypical Contrastive Learning (PCL) [163], or Swapping Assignment Between Views (SwaV) [34]. In [145], the class labels for a supervised learning task are provided as cluster information to improve on the traditional self-supervised instance discrimination task.

3.2.3 A Taxonomy of Encoders

In contrastive representation learning, a learned mapping from inputs to the embedding space needs to satisfy two purposes: mapping to a general and powerful representation of the input data, and an efficient and effective embedding that allows measurement of the distances between samples. We divide the model in our contrastive representation learning framework into two components based on recognising the purpose and functionality of each component i.e. the base encoder and transformation head. The purpose of the encoder is to learn a good mapping from inputs to a general representation space, while the transform heads, depending on the specific choice of similarity, will transform one or multiple representations to a metric embedding for computing a similarity metric. In practice there may be no distinction between the base encoder and the head from a technical point of view as they are just layers of a deep network, stacked on top of each other and jointly optimised through back-propagation with gradient descent but they are functionally distinct, hence the separation.

In this sub-section we focus on a taxonomy of the base encoders. While contrastive learning is general and not restricted to any particular form of encoder, some specific types of encoder and the interactions among them will enable different behaviours for the downstream transform heads and contrastive loss. For each data modality, an appropriate encoder architecture is chosen, so the taxonomy for the encoder will be based on how they are updated with respect to the gradient from the contrastive loss during training.

End-to-end encoders

End-to-end encoders represent the most simple method both conceptually and technically, where the encoders for the queries and keys are updated directly using gradients back-propagated with respect to the contrastive loss function. Since all encoders are updated end-to-end, this can impose a significant requirement on memory. Therefore if the query and keys are of the same data modality, their respective encoders are usually shared with each other so only one copy of the encoder needs to be stored in memory. This way, both the representation for the queries and keys can

be efficiently batch-computed in one single forward pass. However, encoding both the queries and keys end-to-end still requires storing the hidden activations and representation on a Graphical Processing Unit's Video Memory (GPU's VRAM), which will limit the batch size for calculating the contrastive loss.

Online-offline encoders

The online-offline encoder approach alleviates the memory requirement of end-to-end encoders for storing all the queries and keys in a GPU's memory by using an additional offline encoder, which is not updated online by gradient descend directly but updated offline from the online network. In this way, the feature vectors and the hidden activations computed by the offline encoder are not stored on the VRAM. Therefore with this approach, contrastive methods can scale up the number of positive and negative pair comparisons in a batch, independent of the GPU's memory limit.

There are generally two ways to update the offline network, either by using a *past checkpoint* or via a *momentum-based weighted average* mechanism from the online encoder.

Wu et al. [289] decoupled the batch size from the number of negative pairs by storing a detached copy of representations of the entire dataset into a separate *memory bank*. The representations stored in this memory bank are later randomly sampled to serve as the keys, while the queries are encoded by the online network from two different transformations of the same images. The representations computed from the online encoder for the queries are then stored in the memory bank to be used as the keys for the next epoch. This approach effectively uses an online encoder's checkpoint from the previous epoch as the offline encoder for negative keys in the current epoch, with a memory mechanism to avoid redundant computation.

Momentum Contrast (MoCo) [110] further reduces the need to store an offline representation of the entire dataset in the memory bank through the use of a dynamic *memory queue*. The offline momentum encoder is a copy of the online encoder, with parameters being an exponentially-weighted average of that of the online encoder. At every iteration, the latest batch of feature vectors from the momentum encoder are pushed to the memory queue while the oldest batch of features are discarded from the queue. The momentum queue therefore retains a more consistent set of negative keys to the queries and keys encoded online, compared to the memory bank's feature vectors which are only updated once per epoch.

Pre-trained encoders

Another case of not having to keep an encoder in the GPU's memory is when an encoder is already pre-trained and does not need to be updated at all. This usually happens in cross-modal learning or in a knowledge distillation setting, where contrastive methods are used to learn a mapping to the same representation space of another encoder. This approach decouples the task of learning representation for each modality and can simplify the learning task of each encoder while still leveraging the information shared from different data modalities.

In [261], Sun *et. al.* used a pre-trained Bidirectional Encoder Representations from Transformers (BERT) [61] to process discrete automatic speech recognition tokens, while training a separate video BERT model to process continuous video features.

In a knowledge distillation setting, a large pre-trained "teacher" network with frozen weights is used to encode the keys, while a smaller "student" network tries to match the query representation to positive keys from the teacher network. This is a special case where even though the query and key are of the same modality, they are encoded using different encoders. Contrastive Representation Distillation (CDR) [269] uses a large, pre-trained teacher network as the encoder for both the positive and negative keys, while the queries are encoded by a small network learned to match the representation of the teacher network.

3.2.4 A Taxonomy of Transform Heads

The distinction between the base encoder and the transform heads is to separate the ultimate goals of learning a good representation from that of learning an embedding that is efficient and effective for computing and maximising the similarity metric. Entangling the main task of learning a representation and the pretext task of learning a similarity metric can leads to unwanted results, such as by only focusing on maximising the similarity between positive samples, the representation is forced to discard potentially useful information. The introduction of an explicit transform head above encoders is a recent development in contrastive representation learning. Prior to the introduction of the transform heads, many methods trained a standard encoder and then performed a comparison of which layers are best suited to use as representation for transfer learning to some downstream tasks. The result was that for most tasks, one of the hidden layers gave the best performance when using a representation for transfer learning or fine-tuning with a downstream classifier.

With the separation from the base encoder and transform heads, it is now also possible to train the same representation from the base encoder with multiple transform heads for different contrastive objectives.

Depending on the specific choice of data similarity (see Section 3.2.2) and its purpose, we categorise transform heads into three types namely *projection*, *contextualisation* and *quantisation* heads which we now describe in turn.

Projection heads

While the representations (the output of encoders) are of a lower dimensionality to the input dimensions, it can still take a relatively large computational effort to measure the similarity distance between representations. The simplest type of transformation serves as a bridge between different vector spaces. These projections can be a simple linear transformation or a non-linear MLP. With the projection head, the dimensionality for the representation $\mathbf{v}$ can be larger than the dimensionality of the metric embedding $\mathbf{z}$, so that more information can be retained in the representation while also allowing for efficient computation of the similarity metric in the space of $\mathcal{Z}$.

The early contrastive methods that report transfer learning results from the best hidden layers are effectively using the base of the network as a feature encoder and the top of the network as the non-linear projection head. In more recent work, [300] explicitly uses a linear and [41] uses a non-linear 2-layer MLP as the projection head after the base encoder.

Instead of projecting the representation of the query and key encoders to a common metric space, a transformation head can also be used to bridge directly from one metric space to another. In [100], in addition to a projection head from representation space to metric space, an additional "prediction" network projects the metric embedding of an online network to the the metric embedding of an offline encoder.

A common challenge in many representation learning methods is the problem of "dimensionality collapse", where the learned representation only spans a smaller subspace than its given capacity. Based on the theoretical work by Jing et al. [135], it has been shown that the projection head also plays an important role in preventing dimensionality collapses in common constrastive learning methods.

Contextualisation heads

In some settings, the projection heads can be more elaborate than just simply projecting the representation down to a lower dimension. For the task that defines similarity based on the context-instance relationship (Section 3.2.2), a special kind of transform head is needed to aggregate multiple feature vectors into a contextualised embedding.

In *Contrastive Predictive Coding* (CPC) [204] where similarity is defined from

the past-present relationship, a GRU [46] head is applied over previous time steps to aggregate the past information into a contextualised embedding. This is equivalent to an ordered autoregressive head that forces the head to learn generalisable features that are informative when predicting the correct future separate from the incorrect future.

In Deep InfoMax (DIM) [118], where global features are compared with local information in the feature maps, convolution layers with pooling are used to aggregate the feature maps into one single global vector. Similar to DIM, in InfoGraph [262] where contrastive learning is applied on a graph network, a transform function summarises all the patch representations into a single fixed length graph-level representation.

For models based on the Vision Transformer architecture, there is no explicit head for the context information. Instead, extra tokens are added to the input patches and the global information is jointly learned and stored throughout the network layers. This context token is usually denoted *CLS* short for "class" since it originally was designed to support classification tasks.

As distinct from the projection head where the representation is only projected down, the contextualised metric embedding $\mathbf{z}$ serves a different function and holds different kinds of information. Depending on the downstream task where the contextual information is helpful or not, the contextualised embedding $\mathbf{z}$ can actually be used instead of, or in conjunction with, the representation embedding $\mathbf{v}$.

Quantisation heads

While a contextualisation head aggregates multiple representations together, a quantisation head is the opposite in that it reduces the complexity of the representation space by mapping multiple representations into the same representation.

For example, wav2vec 2.0 [13] uses a Gumbel-softmax [130] quantisation head to map the continuous audio signal into a discrete set of latent vectors (i.e "code book").

In methods that combine contrastive learning with clustering approaches such as SwAV [34], a Sinkhorn-Knopp algorithm [52] is used as a quantisation head in order to map a representation of individual samples into a soft cluster assignment vector.

3.2.5 A Taxonomy of Contrastive Loss Functions

Contrastive loss is one of the key differences between contrastive methods and other representation learning approaches. The most prominent difference is that in the contrastive loss formulation, the target can be dynamically defined in terms of the

metric embedding instead of having fixed targets. While most discriminative models measure loss with respect to a prediction label for example using class labels, and generative models measure loss in the input space (e.g. reconstruction loss), contrastive losses measure the distance, or similarity, between embeddings in the latent space.

All forms of contrastive losses can be generally decomposed into two components: a *scoring function* that measures the compatibility between two vectors and the actual *form* of the loss that enforces minimisation and maximisation given a set of query and key vectors.

Minimising the distance between samples is the ultimate goal of any contrastive loss function. However naively minimising the distances between positive pairs can lead to a catastrophic collapse, e.g. the distances between any pairs can be reduced to zero by making the model $f(\mathbf{x};\theta)$ constant with respect to any input $\mathbf{x}$. To prevent this collapse from happening, the contrastive loss function can explicitly use negative pairs that are forced to have a large distance in the embedding space, or we can implicitly employ other assumptions and architecture constraints. For example, in some recent work such as BYOL [100] or [74], negative pairs are not employed explicitly, and here the authors do not refer to their method as a "contrastive learning" approach. However, we consider all methods that contrast between a query and positive keys to learn similarity as contrastive learning methods, regardless of whether explicit negative pairs or architectural constraints are used to prevent the representation from collapsing.

Given the goal of optimising the distance or similarity score, contrastive loss functions can generally be classified based on their motivation and the specific form of how they are formulated. Below we will discuss the different types of scoring functions and then look at the three major forms of contrastive loss functions.

Scoring functions

The scoring function measures compatibility between two vectors either in terms of *similarity* or *distance*. Depending on the specific loss function, for positive pairs either the similarity score is maximised or the distance metric is minimised.

For contrastive losses that operate on the distance notion, usually a simple Manhattan or Euclidean distance (also known as L1 and L2-norm distance) $D(\mathbf{q},\mathbf{k}) = \|\mathbf{q}-\mathbf{k}\|_2$ is used. Distance-based scoring function are often used in energy-based hinge loss functions (Section 3.2.5).

The softmax-based loss requires computing the normalisation term in the denominator, which requires global communication between all the samples. This is especially costly when trying to scale contrastive methods to larger batch sizes. SigLIP [307] proposed to simplify the contrastive loss with a sigmoid loss in place

of softmax, effectively turning the problem into a binary classification, similar to earlier Noise-contrastive estimation methods. Combining this with an efficient implementation in a large scale distributed training setup, they were able to train an image-language contrastive model with batch size of up to 1 million samples.

On the other hand, scoring functions can measure similarity via a simple dot product $S(\mathbf{q}, \mathbf{k}) = \mathbf{q}^\top \mathbf{k}$ between two vectors. The range of similarity scores in this case is unbounded and dependent on both the orientation and magnitudes of the vectors in the sub-space. Since similarity can be made arbitrarily large by increasing the magnitude, one possible solution is to include a normalisation term for the vector's magnitude $\|\mathbf{z}\|^2$ in the final loss function, as is done in [289]. Another method to get rid of dependency on magnitude is to use the cosine similarity, which is computed as the dot product between two unit vectors $S(\mathbf{q}, \mathbf{k}) = \frac{\mathbf{q}^\top \mathbf{k}}{\|\mathbf{q}\|\|\mathbf{k}\|}$. The cosine similarity is bounded between -1 and 1 for anti-parallel and parallel vectors respectively, and equal to 0 for orthogonal vectors. This is most commonly used as a scoring function in modern contrastive loss functions such as the NT-Xent loss in SimCLR [41]. In this way, the representations can still has arbitrary angles and length, while the contrastive only concerns with the angles between the metrics embeddings. Another popular option to measure similarity is the bi-linear model $S(\mathbf{q}, \mathbf{k}) = \mathbf{q}^\top \mathbf{A} \mathbf{k}$, in which the matrix $\mathbf{A}$ is learned and can be considered as a linear projection from the sub-space of $\mathbf{q}$ to sub-space of $\mathbf{k}$, before the dot product operation is performed. The original InfoNCE loss [204] uses this bi-linear model as the scoring function.

In the extreme case, the scoring can also be a learned module and be optimised together with the other modules during training, similar to the discriminator network of a GAN [87]. Different from a GAN's discriminator that evaluates one sample at a time, the learned scoring function concatenates multiple metric vectors together as input and measures the correspondence between them. Though it might be thought that a learnable module is better than a hand-crafted scoring function, using a neural network as a scoring function come with disadvantages. The learned discriminator takes up computational resources that are potentially more helpful for the feature encoder. Therefore, a powerful discriminator can make up for poor representation extracted from an encoder by focusing on learning a good discriminator for bad a representation vector instead of learning a useful representation in itself. The learned scoring functions are also often based on the classification objective, whether the two inputs are compatible or not [6]. It does not provide an explicit measurement of distance and similarity in the latent space, which many downstream applications rely on. Therefore in this thesis, we mostly focus on methods that uses a contrastive loss with relatively simple scoring functions.

Energy-based margin losses

Energy-based Model (EBM)[160] are a general class of models that associate an energy (distance score) with each configuration of the variables to be modelled (pairs of query and keys vectors). Training an EBM involves associating a low energy (small distance) to desired configurations of the variable (positive pairs) and high energy to undesired configurations of variables (negative pairs). Unlike a properly normalised probabilistic model, making the energy for one particular configuration low does not necessarily make energy for other configurations higher. That is why most energy-based models must employ explicit negative comparisons in computing the total loss.

Motivated from EBM, Chopra, Hadsell, and LeCun [47] first introduced and then reformulated in [105] the original "contrastive loss" that uses Euclidean distance $D(\mathbf{q}, \mathbf{k}) = \|\mathbf{q} - \mathbf{k}\|_2$ as the scoring function in the embedding space. To avoid confusion with the general class of all contrastive loss functions, we will refer to this as the "pair loss". The pair loss operates on a pair of query and key, where distance between positive pairs is minimised while the distance between negative pairs should be larger than a given margin, and formally takes the form:

$$\mathcal{L}_{\text{pair}} = \begin{cases} D(\mathbf{q}, \mathbf{k})^2, & \text{if } \mathbf{k} \sim p^+(\cdot|\mathbf{q}) \\ \max(0, m - D(\mathbf{q}, \mathbf{k})^2), & \text{if } k \sim p^-(\cdot|\mathbf{q}) \end{cases} \tag{3.2}$$

where the margin $m > 0$ acts as a radius around the query, for which only negative keys $\mathbf{k}^-$ within this radius are pushed away from $\mathbf{q}$ and contribute to the total loss value.

While the *pair loss* only requires the distance of negative pairs to be larger than a fixed margin, the *triplet loss* [285, 49, 37] enforces the *relative* distance between positive and negative pairs given in a triplet of *(query, positive key, negative key)*:

$$\mathcal{L}(\mathbf{q}, \mathbf{k}^+, \mathbf{k}^-) = \max(0, D(\mathbf{q}, \mathbf{k}^+)^2 - D(\mathbf{q}, \mathbf{k}^-)^2 + m) \tag{3.3}$$

While conceptually simple and widely adopted in multiple metric learning applications [276, 121, 239], the pair and triplet losses usually suffer from slow convergence because of the limited interactions between samples. In pair loss, only one comparison to either a positive or negative key is computed for a given query, while triplet loss simultaneously compares the relative distance from a query to one positive and negative key. Mining techniques to find "hard" negative samples to avoid easy pairs that provide no substantial learning signal are essential components of these learning systems. To increase the number of interactions for a query, methods such as *Lifted Embedding loss* [252] and a generalised version of it [113] improved

on the margin formulation of triplet loss to take into consideration multiple positive and negative keys for a query within a batch.

Probabilistic NCE-based losses

A form of contrastive loss can also be motivated from the probabilistic softmax classification problem. Consider the traditional supervised parametric softmax classification objective, the probability that a query is correctly recognised as belonging to the i-th class among n classes is

$$p(i|\mathbf{q}) = \frac{\exp(\mathbf{q}^\top \mathbf{w}_i)}{\sum_{j=1}^{n} \exp(\mathbf{q}^\top \mathbf{w}_j)} \tag{3.4}$$

where $\mathbf{w}_j$ is a vector specific to the class i in the data set. This vector $\mathbf{w}$ in the parametric formulation of softmax serves as a class prototype and does not allow explicit comparison between representations.

Motivated by this, a non-parametric version for the softmax function that correctly identifies the positive for a given query from a set $\mathcal{K}$ and contains all negative keys with one positive key can be defined as follows:

$$p(\mathbf{k}^+|\mathbf{q}) = \frac{\exp(\mathbf{q}^\top \mathbf{k}^+)}{\sum_{k \in \mathcal{K}} \exp(\mathbf{q}^\top \mathbf{k})} = \frac{\exp(\mathbf{q}^\top \mathbf{k}^+)}{Z(\mathbf{q})} \tag{3.5}$$

with $Z(\mathbf{q})$ as the normalising constant, or partition function for a given query.

The learning objective is then to maximise the joint probability or equivalently to minimise the negative log-likelihood over the training set:

$$\mathcal{L}(\mathbf{q}, \mathcal{K}) = -\log p(\mathbf{k}^+|\mathbf{q}) \tag{3.6}$$

The normalisation constant $Z(\mathbf{q})$ in the denominator of the non-parametric softmax in (3.5) is expensive to evaluate because it needs to sum over all the negative keys in the dataset for a given query. Noise Contrastive Estimation (NCE) [103, 104] is an estimation method for an unnormalised probabilistic model that avoids the need to evaluate the partition function through a proxy binary classification task, where the binary task is to discriminate between *data samples* (positive keys) and the *noise sample* (negative keys).

Following the original NCE formulation and assuming a uniform *noise distribution* of negative samples $p^-(\cdot|q) = 1/n$ and that we sample noise negative keys m times more frequently than the positive key, the posterior probability of the pair

$(\mathbf{q}, \mathbf{k})$ sampled from the positive distribution $p^+(\cdot, \cdot)$ (denoted by $D = 1$) is:

$$p(D = 1|\mathbf{q}, \mathbf{k}) = \frac{p(\mathbf{k}^+|\mathbf{q})}{p(\mathbf{k}^+|\mathbf{q}) + m \cdot p(\mathbf{k}^-|\mathbf{q})} \tag{3.7}$$

With $p(D = 1|\mathbf{q}, \mathbf{k}) = \frac{1}{1+\exp(S(\mathbf{q},\mathbf{k}))}$ parametrised by a sigmoid function with the similarity scoring function $S(\mathbf{q}, \mathbf{k})$, the approximated NCE binary training objective then becomes:

$$\mathcal{L}_{NCE-binary}(\mathbf{q}, \mathcal{K}) = -\mathbb{E}_{p^+}[\log p(D = 1|\mathbf{q}, \mathbf{k})] \\ - \mathbb{E}_{p^-}[\log(1 - p(D = 1|\mathbf{q}, \mathbf{k})] \tag{3.8}$$

This NCE objective has been used widely in learning language models [192] and word embeddings [191]. A slightly different variation of binary NCE is Negative Sampling (NEG) [186] which focuses on learning good word embeddings.

Instead of having a binary task that decides whether each key is positive or negative, suppose we want to correctly identify and rank the positive key with highest similarity to the query in a set $\mathcal{K} = \{\mathbf{k}^+, \mathbf{k}_1^-, ..., \mathbf{k}_n^-\}$ with one positive key and n negative keys. Jozefowicz et al. [137] extended the *local* view of binary NCE to a *global* or *ranking* view, such that the conditional distribution of key at index i is the positive key is given by:

$$p(i|\mathbf{q}, \mathcal{K}) = \frac{p^+(\mathbf{k}_i|\mathbf{q})\Pi_{j\neg i}p^-(\mathbf{k}_j|\mathbf{q})}{\sum_{n=1}^{N} p^+(\mathbf{k}_n|\mathbf{q})\Pi_{j\neg n}p^-(\mathbf{k}_j|\mathbf{q})} \tag{3.9}$$

If we let $p(i|\mathbf{q}, \mathcal{K}) = \frac{\exp(S(\mathbf{q},\mathbf{k_i}))}{\sum_j^N \exp(S(\mathbf{q},\mathbf{k_j}))}$ be parameterised by a softmax function, the approximated global ranking NCE training objective then becomes:

$$\mathcal{L}_{NCE-global}(q, \mathcal{K}) = \mathbb{E}_{P(i|q,\mathcal{K})}\left[-\log\frac{\exp(S(q, k^+))}{\sum_{k\in\mathcal{K}}\exp S(q, k)}\right] \tag{3.10}$$

The reader is referred to [176, 260] for more detailed treatment of different variations of NCE-based objectives.

Sharing the same motivation with the *Lifted Embedding loss* from the metric learning objective instead of by the NCE objective, Sohn [251] independently proposed the *Multi-class n-pair* loss that has the same formulation as the NCE-global objective in Eq. (3.10) and uses samples in the same mini-batch as the negative samples to save memory during computation. By formulating it as a multi-class classification problem, this loss automatically incorporates multiple negative keys for comparison, and is thus very effective.

In more recent work, a slightly different form of this loss called the Normalised Temperature Cross Entropy (NT-Xent) [41] loss with a temperature parameter τ to

control sensitivity of the cosine similarity scoring function is used

$$\mathcal{L}_{NT-Xent}(\mathbf{q}, \mathcal{K}) = -\log \frac{\exp(\frac{\mathbf{q}^\top \mathbf{k}^+}{\|\mathbf{q}\|\|\mathbf{k}^+\|\tau})}{\sum_{k \in \mathcal{K}} \exp(\frac{\mathbf{q}^\top \mathbf{k}}{\|\mathbf{q}\|\|\mathbf{k}\|\tau})} \tag{3.11}$$

The temperature τ has the same effect of controlling the attraction-repulsion radius around the query, similar to the margin m in the margin-based contrastive loss in Section 3.2.5.

Mutual information-based losses

Mutual Information (MI) has a long history in representation learning for various methods that aim to maximise the MI a representation $\mathbf{z}$ and its inputs $\mathbf{x}$. In the same spirit, contrastive learning methods motivated from MI aim to learn a mapping that maximise the mutual information between representations of different views of the same scene, which is upper bounded by the MI between the representation and the input of a scene.

Oord, Li, and Vinyals [204] first proved that minimising the InfoNCE loss based on NCE is equivalent to maximising a lower bound on the MI. Inspired from NCE, InfoNCE comes to the same formulation of the classification-based *N-pair loss* in Eq. (3.10), and shows that minimising this loss also maximises a lower bound on the mutual information between the input and the representation. Having the same form as the multi-class n-pair loss [251] and NT-Xent [41] but using a bi-linear layer as a scoring function instead of a dot product, this form of contrastive loss is currently the most popular due to its effectiveness and simplicity in implementation, as well as a theoretical guarantee based on MI.

Proposed independently of InfoNCE, DIM [118] also formulated the contrastive learning problem as MI maximisation and evaluated different MI estimators, such as the Donsker-Varadhan [66], the Jensen-Shannon estimator [201] and the InfoNCE [204].

Some recent work [217] performed a review of different MI estimators and derived a new continuum of multi-sample lower bounds that describes the bias-variance and efficiency-accuracy tradeoffs, as well as showing the generalisation bound of MI in the context of contrastive learning.

However, even though mutual information is a principled motivation for contrastive losses based on the information bottleneck principle, simply maximising the mutual information in positive pairs does not guarantee a successful application of the contrastive loss concept. Tschannen et al. [271] argue and provide empirical evidences that the success of contrastive losses can not be attributed to mutual information alone.

3.3 Development of Contrastive Learning

Now we will briefly examine the major developments in contrastive methods over time, that span over multiple sub-fields and domains.

The core idea of learning by comparing between separate but related data points, without any supervised signal, dates back to 1992 to work by Becker and Hinton [19] and by Bromley et al. [26] in 1993. While Becker and Hinton [19] formulate the problem as learning invariant representations by maximising mutual information among different views of the same scene, Bromley et al. [26] introduces the "Siamese Network" composed of two identical weight-sharing networks in a metric learning setup. These are the first examples of the general principle of learning by directly comparing between different training samples.

In 2005, Chopra, Hadsell, and LeCun [47] [105] created the foundation for the contrastive learning framework with the original contrastive pair loss for discriminative models to learn an invariant mapping for recognition and verification problems. Instead of having to define non-linear similarity relationships using some simple metric in the input space, the contrastive pair loss demonstrates the ability to learn a representation space in which a simple distance metric in the embedding space approximates a notion of similarity in the input space.

Inspired by a form of triplet loss used in [285], Collobert and Weston [49] trained an unsupervised language model, and Chechik et al. [37] learned an image similarity model using a ranking triplet loss. Later, the triplet loss was applied in the context of a deep neural network and has been shown to be capable of learning fine-grained image similarity [276], or a useful representation [121].

To address the limitations of slow convergence and instability of the pair and triplet contrastive losses, Song et al. [252] and Sohn [251] proposed loss functions that improve the number of comparisons for a query in an iteration. While using hard negative and positive samples has been a common component in successfully applying contrastive methods, Manmatha et al. [179] and Hermans, Beyer, and Leibe [113] argue for the case that quality of data pairs used in training are also of paramount importance for pair and triplet losses in the metric learning setting.

While there have been approaches to using probabilistic approaches to learning metric embeddings [267], most successful applications up to now all use the energy-based pair or triplet loss due to the computational requirements to compute the normalisation constant in probabilistic loss. In 2010, Gutmann and Hyvärinen [103] introduced Noise Contrastive Estimation (NCE), a simple conceptual strategy for estimating an unnormalised statistical model by contrasting between the data and noise distributions.

In natural language processing that processes discrete input text tokens, this

Table 3.1: A table summary of the development of contrastive learning methods. Entries are sorted in chronological order of first disclosure. The topics of contribution include *foundational* ideas behind contrastive learning, the development for different forms of the contrastive *loss*, how *similarity* is defined and new *applications* of contrastive learning methods.

Paper	**Short description**	**Contribution**
Becker and Hinton [19]	Maximise MI between two views	Foundation
Bromley et al. [26]	Siamese network in metric learning setting	Foundation
Chopra, Hadsell, and LeCun [47]	Learn similarity metric with contrastive pair loss	Energy-based loss, Application
Hadsell, Chopra, and LeCun [105]	Learn invariant representation from pair loss	Energy-based loss, Application
Weinberger, Blitzer, and Saul [285]	Learn distance metric with triplet loss	Energy-based loss
Collobert and Weston [49]	Learn language model with triplet loss	Application
Chechik et al. [37]	Learn image retrieval model with triplet loss	Application
Noise Contrastive Estimation [103]	Introduce NCE, a general methods to learn unnormalised probabilistic model	Probabilistic loss
Mnih and Teh [192]	Learn language model with NCE-based loss	Application
Mikolov et al. [186]	Learn word embedding with Negative Sampling (NEG), a modified version of NCE	Probabilistic loss, Application
Wang et al. [276]	Learn fine-grained image similarity using deep network and triplet loss	Application
Wang and Gupta [278]	Use video's sequential coherence to learn unsupervised video representation	Similarity, Application
Lifted-structure loss [252]	Extend triplet loss to multiple positive and negative pairs per query	Energy-based loss
N-pair loss [251]	Proposed non-parametric classification loss with multiple negative pairs per query	Probabilistic loss
Manmatha et al. [179]	Focus on the quality of negative samples through a distance-weighted margin loss	Similarity, Energy-based loss
Hermans, Beyer, and Leibe [113]	State the important of mining hard samples in triplet loss	Similarity
Wu et al. [289]	Self-supervised representation with instance discrimination Memory bank to holds keys for next epoch	Application Encoder
CPC [204]	Mutual Information with the contrastive loss Define similarity with past-future context-instance relationship	Mutual Information loss Similarity
DIM [118]	Evaluate multiple mutual information bound for the contrastive loss Global-local context-instance relationship	Mutual Information Loss Similarity
MoCo [110]	Use momentum encoder to store features to memory queue	Encoder
SimCLR [41]	Simplify and demonstrate large empirical improvement in instance discrimination task Focus on the use of separate heads	Application Transform heads
BYOL [100]	Learning similarity without negative samples	Loss

form of NCE-based contrastive loss has been used to train powerful language models [192] or to learn useful word embeddings [191, 186] from a large unlabelled corpus of text.

Also motivated from the mutual information maximisation perspective similar to [19], in 2018 CPC [204] and DIM [118] made the connection between minimising a contrastive loss with maximising a lower bound of the mutual information between different views.

The instance discrimination task that drove the progress of contrastive methods in the past few years is introduced in [289]. Simplifying the framework for instance discrimination and focusing on learning representations with only augmentation methods, Ye et al. [300] and Misra and Maaten [190] showed that pre-training with contrastive loss can outperform supervised-only training for a computer vision task. To achieve the best results with contrastive loss, training with large batch sizes on a large GPU cluster is required. Methods such as Momentum Contrast (MoCo) [110] were introduced to reduce the requirement for large batch sizes. Using an online and momentum-updated offline network, MoCo proposed to view contrastive learning as a form of dictionary lookup and raised questions around how best to retain consistency between offline and online networks to perform similarity matching between the queries and keys.

Using extra network heads on top of the learned representation has been used previously, but it was mostly out of necessity, for example to aggregate context information from multiple time steps such as in CPC [204]. SimCLR [41] proposed an explicit projection head to separate between the tasks of learning a representation and optimising for the contrastive objective. This distinction raises the question of what are the optimal design choices for the base encoder and representations for recent work such as SimCLRv2 [43]. This separation enabled other work to use multiple heads and contrastive objectives when optimising for the same underlying representation [74, 292].

Local aggregation [314] spearheaded the direction of combining clustering methods with instance discrimination contrastive learning, while in [277, 100, 74] the authors raised the question of whether negative samples are necessary at all where they propose a different contrastive loss function to avoid the collapse of the representation with additional implicit constraints.

Table 3.1 provides a brief summary of some prominent papers over the development of contrastive learning.

3.4 Applications

We now look at various data domains and problem topics to which contrastive learning representations have been applied. This is done through the lens of the generalised Contrastive Representation Learning framework introduced in Section 3.2.

3.4.1 Language

Following the idea proposed in [203] to learn a language model discriminatively, Collobert and Weston [49] learned a language model to perform a two-class classification task to determine whether and how the middle word of a context window is related to its context or not. They used positive examples as instances of such word triples taken from Wikipedia and created negative examples by replacing the middle word in a triplet by a random word and trained the model with a triplet loss.

Later, Mnih and Teh [192] adapted NCE [103] and proposed a more efficient algorithm to learn a language model using a probabilistic contrastive loss, where the context query includes all the previous words, the positive key is the next word in a sequence and the negative keys are sampled from a unigram distribution of words in the corpus.

With the introduction of the Skip-gram and CBOW algorithms [186] to learn word representations which depend heavily on the tree structure of the hierarchical softmax, Mnih and Kavukcuoglu [191] used NCE to avoid having to compute the normalisation term of the softmax. Also inspired by NCE, Mikolov et al. [187] proposed a slightly different method called Negative Sampling (NEG) that focuses solely on learning good word representations with the trade-off of losing the probabilistic properties from NCE.

Recently, the Bidirectional Encoder Representation from Transformer (BERT) [61] model learns bidirectional word representations using the Transformer architecture's decoder [272] and demonstrated great performance for transfer learning in multiple downstream tasks. XLNet [299] modified BERT's masked language model objective to include an autoregressive objective. While these language model objectives are usually referred to as a form of denoising autoencoder that try to reconstruct the original input, in the case of learning word embeddings which is just a lookup layer from index to vector, there is no difference between reconstructing and contrasting between feature vectors and thus this work does fall under the remit of being a form of contrastive learning.

Under the mutual information maximisation framework, Kong et al. [155] showed that BERT or XLnet also maximise global-local mutual information, whereas the next sentence prediction pre-training task can be seen as constructing similarity pairs using the sequential coherence property. With this insight, Kong et al. [155] also

Table 3.2: A summary of methods that applied contrastive methods on language data. The color for defining similarity in query and keys encodes: Multi-sensory, Data transformation, Context-Instance, Sequential Coherence, Clustering. Colors for encoder represent: End-to-end, Online-Offline, Pre-trained. Colors for transform head represent: Projection, Contextualisation and Quantisation.

Method	**Query**	**Positive keys**	**Negative keys**	**Encoder**	**Transform head(s)**	**Loss**
Collobert and Weston [49]	Surrounding words	Centre word	Random words	Embedding	Max-pooling	Triplet loss
Mnih and Teh [192]	Previous words	Next word	From unigram distribution	Bi-linear	Position-dependent weighting	Binary NCE
word2vec [191]	Surrounding words	Center word	From unigram distribution	Vector Bi-linear	Position-dependent weighting	Binary NCE
word2vec [186]	Surrounding words	Centre word	From modified unigram distribution	Vector Bi-linear	Position-independent weighting	NEG
QuickThought [171]	Surrounding sentences	Centre sentence	Sentences outside windows	GRU	No	NEG
CPC [204]	Past sentences	Next sentences	Random sentences	ConvNet	GRU	InfoNCE
Bert-NCE [45]	Masked sentence	Masked word	Random words	Embedding	Transformer	Binary NCE
Sentence-Bert [225]	Query sentence	Same-paragraph sentences	Random sentences	Transformer	Pooling	Triplet loss
InfoWord [45]	Sentences with masked n-gram	Original masked n-gram	Random n-gram	Transformer	No	InfoNCE

proposed BERT-NCE, a variant of BERT that uses an NCE-based loss instead of the full softmax over the entire vocabulary, making it more aligned with contrastive learning methods. Inspired by DIM [118], they also introduce InfoWord that aims to maximise the mutual information between local and global representations of a sentence. The queries for the global representation are the sentence with a contiguous masked chunk which is an n-gram, the positive keys are the local representation of the original n-gram while negative keys are randomly sampled n-grams. The final model used InfoNCE loss to minimise the mutual information lower-bound for both the masked language model and the global-local representation objective.

In learning representations for units larger than words, Quick-Thought [171] extends the Skip-gram model for word embedding to learn representations for entire sentences. A GRU [46] encodes word-by-word a query sentence and a nearby sentence as the positive keys, while the negative keys are encoded from sentences outside the context window. The final hidden state of the GRU is treated as the sentence embedding.

CPC is a general contrastive learning method that can be applied to many different data modalities. For text data, CPC encodes the context query using past sentences with the positive keys as the future sentence. A 1-D convolution network is used as the encoder to encode the entire sentence, while a GRU acts as a context head and aggregates information from past sentences to predict the representation of future sentences.

SentenceBERT [225] extended word representations from BERT to explicitly learn a sentence embedding using the triplet loss. Two sentences from the same paragraph are considered positive pairs and are negative otherwise. After obtaining individual word representations from BERT, either the special token CLS or a pooling operation is used over the entire sentence to obtain the sentence representation.

Inspired from the success of data transformation-based contrastive methods in computer vision, Fang et al. [77] extended this idea and introduces CERT to learn sentence-level representations. To create positive pairs of sentences, CERT creates two different sentences which are similar in meaning by back-translating, using a machine translation model to translate a sentence into a target language and using another translation model to convert it back to the source language. CERT uses BERT as its encoder and uses InfoNCE as the contrastive loss function.

As yet another alternative approach, Chi et al. [45] used contrastive methods to learn cross-lingual sentence representations using a parallel corpus. In InfoXML, the objective includes a combination of maximising monolingual and cross-lingual token-sequence (global-local) information, and cross-lingual sentence-sentence (multiview) information. The CLS token from the base BERT encoder is used as the sentence representation with a linear projection head. A momentum encoder is used to encode

the query while the online encoder is updated using the InfoNCE loss.

Not limited to natural language but still a form of language, [128] learns a functional-equivalent of program code representation by generating similar code snippets using different augmentation techniques from the compiler literature. The transformer's representation of each token is averaged to obtained the representation for the entire program and InfoNCE is used as the contrastive loss.

A summary of the methods that learn language representations using Contrastive learning is shown in Table 3.2.

3.4.2 Vision

Motivated by the challenges of recognition, verification and fine-grained classification problems, Chopra, Hadsell, and LeCun [47] introduced the contrastive pair loss function in the context of metric learning. Such applications need to deal with data with high intra-class variance (e.g same face but different lighting condition and angles) and low inter-class variance (e.g different faces but taken by the same camera setup). The explicit formulation of a contrastive learning objective to minimise the distance between inputs of the same class whilst maximising the distance between inputs of different classes is a direct attempt to solve this problem. On the other hand, Hadsell, Chopra, and LeCun [105] demonstrated that the contrastive loss will learn an invariant mapping for many irrelevant input features in order to be able to map different inputs to the same neighbourhood in the embedding space.

Building on the intuition of invariant mapping and its application in metric learning, Chechik et al. [37] learned a large scale image similarity model for retrieval using the triplet loss.

Moving beyond metric learning applications, Hoffer and Ailon [121] used a similar triplet architecture but focused on learning image representations simply from using the class labels to denote similar pairs. Wang and Gupta [278] extended this idea beyond supervised learning by learning visual representations from video with the help of an unsupervised tracking method. The corresponding patches provided by the tracker are used as the positive pairs while the hard negative pairs are mined from elsewhere in the dataset.

Among the first to exploit sequential coherence for defining triplets, Sermanet et al. [242] introduced the Time-Contrastive Network (TCN), a self-supervised method to learn a view-agnostic but time-sensitive representation from unlabelled videos. Two simultaneous views from different cameras, or two consecutive frames from the same view are defined to be similar, while two frames far apart in time but from the same camera view are defined to be dissimilar.

Recently contrastive learning has received a lot of attention due to its success-

Table 3.3: A summary of methods that applied contrastive methods on vision data. The color for defining similarity in query and keys encodes: Multi-sensory, Data transformation, Context-Instance, Sequential Coherence, Clustering. Colors for encoder represent: End-to-end, Online-Offline, Pre-trained. Colors for transform head represent: Projection, Contextualisation and Quantisation.

Method	**Query**	**Positive keys**	**Negative keys**	**Encoder**	**Transform head(s)**	**Loss**
Chopra, Hadsell, and LeCun [47]	Face query	Same face from different camera	Different faces	ConvNet	None	Pair loss
DrLIM [105]	MNIST digit	Same digit shifted	Random MNIST digits	Convolution	None	Pair loss
Chechik et al. [37]	Query image	Same label images	Different labels images	Bag-of-local-descriptors	None	Triplet loss
Hoffer and Ailon [121]	Query image	Same label images	Random images	Multi-scale ConvNet	None	Triplet loss
Wang and Gupta [278]	Patch from first frame	Tracked patch from last frame	Random sampling and hard negative mining	ConvNet	None	Triplet loss with cosine distance
TCN [242]	Frame t from camera 1	Frame t from camera 2	Frame $t+k$ from camera 1	ConvNet	None	Triplet loss
Wu et al. [289]	Augmented image	Same image from memory bank	Random images from memory bank	Convolutional Net + Memory Bank	None	Non-parametric classification
Ye et al. [300]	Augmented image	Differently augmented same image	Random images from batch	ConvNet	Linear	Binary NCE
PIRL [190]	Augmented image	Augmented patches of the same image	Random images	ConvNet + Memory bank	Query image: Linear layer, key patches: Pooling	Binary NCE
MoCo [110]	Augmented image	Augmented query image	Random images from momentum queue	Convolutional Net + Momentum Encoder	Linear	InfoNCE
SimCLR [41]	Augmented image	Augmented query image	Random images from batch	Convolutional Net	MLP	NT-Xent
CPC [204]	Aggregated patch's features	Subsequent patches	Random patches	Convolution	Convolutional row-GRU	InfoNCE

Method	Query	Positive keys	Negative keys	Encoder	Transform head(s)	Loss
DIM [118]	Global feature	Local feature maps of query	Local feature maps of random images	Convolution Net	Convolution Net + Pooling	InfoNCE
ST-DIM [4]	Global feature at time step t	Local feature map at time step $t+1$	Local feature map at random time step t^*	Convolution Net	MLP	InfoNCE
AMDIM [11]	Augmented global feature	Augmented multi-scale local features of query	Multi-scale feature map of random images	Convolution Net	Convolutional Net + Pooling	InfoNCE
VINCE [88]	Query frame	Frames from same video	Frames from random videos	Convolutional Net + Momentum Encoder	MLP	InfoNCE (multiple positive pairs)
Local Aggregation [314]	Query image	Close neighbours	Background neighbours	Convolutional Net + Memory bank	Linear	InfoNCE
PCL [163]	Augmented image	Augmented and prototypes vectors of query	Feature and prototypes vectors of random images	Convolutional Net + Momentum encoder	MLP	ProtoNCE (instances + clusters InfoNCE)
SwAV [34]	Prototype of query	Prototype of augmented query	No negative samples	ResNet	MLP + soft cluster assignment	Instances + clusters InfoNCE
InterCLR [293]	Augmented query image	Images with same cluster's pseudo-label	Images with different cluster's pseudo-label	Convolutional Net + Memory bank	MLP	InfoNCE
Khosla et al. [145]	Augmented image	Images with same supervised label	Images with different supervised label	ConvNet	MLP	InfoNCE (multiple positive pairs)
BYOL [100]	Augmented image	Augmented query image	No negative pair	Convolutional Net + Momemtum Encoder	Projection MLP + Prediction MLP	MSE
Whitening MSE [74]	Query image	Augmented version of same image	No negative pairs	ConvNet	MLP	MSE

ful application to self-supervised visual representation learning, especially in the Instance Discrimination task introduced by Wu et al. [289]. Following the idea of treating each instance as its own exemplar class [68], a memory bank mechanism was introduced to store the computed representations for use in future iterations, so that the number of negative samples is decoupled from the batch size. The queries are computed online and contrasted with the keys from the memory bank where the global NCE objective is used to learn to discriminate between features of the same instance or not. Looking at contrastive learning as a dictionary lookup problem, He et al. [110] introduce Momentum Contrast that maintains the offline encoder as an exponentially weighted average of the online encoder where it stores the key representations in a queue, weighting more recent key representations as being more important.

Since the difference between the query and the positive keys in instance discrimination is how they are randomly augmented, multiple works such as *Invariant and Spreading Instance Feature* [300], PIRL [190], SimCLR [41] have focused on engineering strong and varied augmentations to yield better representation from the ImageNet [58] dataset without class labels. These methods have attracted special interest because for the first time they outperform supervised ImageNet classification pre-training on multiple downstream vision tasks. SimCLRv2 [43] performed a comprehensive study of contrastive self-supervised learning in semi-supervised settings where few labels are present, and demonstrated state-of-the-art results by contrastive pre-training in various downstream vision tasks.

In a different direction, Oord, Li, and Vinyals [204] proposed Contrastive Predictive Coding (CPC) to learn invariances between context-instance relationships instead. The predictive coding principle in CPC defines context as the past, and that a good representation of the past will possess a strong predictive capability for instances in the future. The predictive power of a representation is modelled as a contrastive objective that maximises the mutual information between the past context and the future instance through the InfoNCE mutual information lower bound. While the CPC method is general and equally well applicable to multiple data modalities, CPCv2 [111] improved on CPC with some architectural design changes specifically for learning from images and evaluating this on label-efficient fine tuning tasks. Expanding CPC into learning representations from natural videos, Dense Predictive Coding (DPC) [106] contrasts between local patches of the feature maps extracted from the past context with the local patches of the features maps extracted from future instances. DPC employs three kinds of negative samples: the easy negatives come from patches encoded from different videos, the spatial negatives come from the same video but at different spatial locations of the feature maps, and the hard negatives come from the same spatial location but from different time

indexes.

Also learning invariances from context-instance relationship, DIM [118] defined context to be a little more general than CPC. A single vector for each image is used as the global representation, while the feature vectors at each spatial location from the feature map at previous layers are considered local features. DIM enforces the contrastive objective using multiple different mutual information lower-bounds but also found that InfoNCE is the most effective, especially with a large number of negative samples. Combining the context-instance strategy with the temporal coherence property of a video, Anand et al. [4] proposed SpatioTemporal DeepInfoMax (ST-DIM) (ST-DIM) that learns to maximise mutual information between global features of the current frame and local features from the next frames. Finally, Augmented Multiscale DIM [11] combined both the global-local objective from DIM [118] and image data augmentation from the instance discrimination task to learn visual representations.

By exploiting temporal consistency as a natural source of image transformation, Video Noise Contrastive Estimation (VINCE) [88] modified the instance discrimination task where instead of contrasting between two augmented views of the same image, VINCE defined positive pairs as two frames from the same video. An additional benefit of this approach is that different objects that are likely to show up in the same video (e.g dog and cat) are also encouraged to be closer than more random pairs (e.g cat and whale). By combining the image data transformation, temporal coherence between frames and global-local correspondence between features, Video DeepInfoMax (ST-DIM) (VDIM) [117] learned effective spatio-temporal representations for downstream tasks on videos.

Exploiting visual similarity to form natural clusters in the representation space has been used previously to learn unsupervised representations [32]. This objective has been reformulated in the form of a contrastive learning method in [314], where a set of close neighbours is aggregated together from a set of background neighbours. Given a query image, the *background neighbours* are an unbiased sample of nearby points measured with cosine distance in the embedding space. An unsupervised clustering algorithm is applied on the set of background neighbours, where the samples in the cluster that includes the query are the *close neighbours*, which act as the set of positive samples for that query. The embedding is learned iteratively using an NCE loss to classify between close neighbours and background neighbours. In addition to just preserving the local smoothness around each instance in the same cluster, Prototypical Contrastive Learning (PCL) [163] also encoded the higher semantic structure of the data into the embedding through the cluster's centroid. Assuming that each data point is associated with a latent class variable, PCL aims to learn both the class's prototype and optimises for points belonging to a cluster

to stay close together through the Expectation Maximisation (EM) framework. In the E-step, k-clusters are obtained by performing k-means on the features from the momentum encoder and the distance from each point to its cluster's prototype is minimised using the InfoNCE loss in the M-step.

Most clustering-based methods up to now are offline in the sense that they require multiple passes over the data to compute features and perform clustering, but Swapping Assignments between multiple Views of the same image (SwAV) [34] proposed an online clustering method to learn unsupervised visual representations. Combined with data transformation approaches in instance learning, two different augmented views of the same images are encoded into features and the clustering assignment for each of the views is computed from a set of trainable "code" vectors. Similarity is enforced through a "swapped" prediction problem where the feature vectors from one of the views is matched with the cluster's code from the other views. No negative pairs are explicitly used in this method but the representation is prevented from collapsing through the batch-wise online code computations. *InterCLR* [293] also performed mini-batch clustering with a set of learned cluster centroids but instead of using a swapped prediction with no explicit negative samples, they modelled the instance-cluster relationship by assigning a pseudo-label for each instance. Samples that shared pseudo-labels are positive pairs while samples that have different labels are negative pairs. All of these clustering-based contrastive methods in a sense enhance the similarity and dissimilarity in the instance discrimination task through using pseudo-labels derived from clustering techniques.

Most of the methods above focus on the self-supervised paradigm and thus refrain from using human-annotated labels. *Supervised Contrastive Learning* [145] directly used class labels to define similarity, where samples from the same class are positive and samples from different classes are negative samples. This method was shown to be more robust to corruption than using the usual cross-entropy loss with the labels alone.

Most of the work above utilised the NCE objective in one form or another, which will usually benefit with more negative samples. Therefore self-supervised contrastive representation learning methods usually require large batch sizes and longer training times than other supervised or self-supervised methods. The training dynamic of contrastive methods can be dissected into two keys properties [277], *alignment* (closeness) of features from positive pairs and *uniformity* (spreading) of the induced representation on a hypersphere. The *uniformity* explains the role of negative pairs in keeping the representation from collapsing and opens up the research direction of using other methods without negative samples to prevent the representation from collapsing. In *SwAV*, similarity is formulated as a swapped prediction problem between positive pairs while the minibatch clustering methods

implicitly prevent collapse of the representation space by encouraging samples in a batch to be distributed evenly to different clusters. In Bootstrap Your Own Latent (BYOL) [100], the similarity constraint between different views are also enforced through a prediction problem, but from an online network to an offline momentum-updated network. The key insight is that by trying to match the prediction from an online network to a randomly initialised network, the obtained representations are already better than those of the random offline network. By continually improving the offline network through the momentum update, the quality of the representation is bootstrapped from just the random initialised network.

In concurrent work, Ermolov et al. [74] proposed a *Whitening MSE* loss, where again the similarity between augmented instances is enforced through the minimisation of MSE distance in the embedding space, while the *whitening* operation common in many image pre-processing pipelines is applied on the representation in batch. The whitened vectors of all samples in a batch, including positive pairs, become distributed and the MSE objective will pull features of positive pairs closer together i.e. the distance between positive pairs is small while the representation space does not collapse into a single cluster.

Focusing on the data scaling aspect, VITO [212] proposes a contrastive method for distilling knowledge from natural transformation from videos. This yields a significantly more robust representation to transformations and adversarial samples.

A summary of the methods that learn visual representations using Contrastive learning is shown in Table 3.3.

3.4.3 Audio

For audio processing, CPC [204] used a strided convolutional network as the base encoder to map from raw audio signal to the representation $\mathbf{v}$ where a GRU RNN head aggregates the information from all previous timesteps to form a contextualised representation $\mathbf{z}$. This contextualised embedding $\mathbf{z}$ is then used as the query where it is contrasted with a set of representations $\mathbf{v}$ with respect to the true future $\mathbf{v}^+$ from the noise $\mathbf{v}^-$.

Built on top of CPC, wav2vec [238] uses another convolutional network to aggregate context information instead of using a recurrent network for the context head. Moving beyond evaluating on frame-wise phoneme classification in CPC, Schneider Schneider et al. [238] evaluated the learned representation of wav2vec and applied the contrastive pre-trained representation to improve a supervised Automatic Speech Recognition (ASR) system. VQ-wav2vec (Vector-quantised wav2vec) [12] modifies the wav2vec architecture by using an additional quantisation head before the context head. The quantisation head is implemented through a Gumbel-softmax [130]

Table 3.4: A summary of methods that applied contrastive methods on audio data. The color for defining similarity in query and keys encodes: Multi-sensory, Data transformation, Context-Instance, Sequential Coherence, Clustering. Colours for encoder represent: End-to-end, Online-Offline, Pre-trained. Colours for transform head represent: Projection, Contextualisation and Quantisation.

Method	**Query**	**Positive keys**	**Negative keys**	**Encoder**	**Transform head(s)**	**Loss**
CPC [204]	Aggregated past context	Future signal	Random signal from same audio clip	Convolution	GRU	InfoNCE
Wav2vec [238]	Aggregated past context	Future signal	Random signal from same audio clip	Convolution	Convolution	InfoNCE
VQ-Wav2vec [12]	Aggregated past context	Future signal	Random signal from same audio clip	Convolution	Convolution + Gumbel softmax	InfoNCE
Wav2vec 2.0 [13]	Masked bidirectional vector	Masked quantised vectors	Random quantised vectors from same clip	Convolution	Gumbel softmax + Masked Transformer	InfoNCE
Nandan and Vepa [197]	Augmented mel spectrogram	Augmented mel spectrogram	Random Mel spectrograms	Convolution	MLP	InfoNCE

to convert the continuous speech signal **v** into a set of discrete codes **c**. The context head is built on top of these discrete codes to form the query context vector **z**. Similar to CPC and wav2vec, the context vector is then compared with another quantised representation **c** to find the representation of the correct future. The discretised speech representation can then be used directly as a representation for other models that expect discrete input such as BERT [61].

All of these methods above encode context representation using only past-to-present information. Inspired from the success of the bidirectional encoding in the transformer model [272], Wav2vec 2.0 [13] replaces the unidirectional context head from vq-wav2vec [12] with a bidirectional masked Transformer.

In a different direction, Nandan and Vepa [197] learned speech representation from audio in mel spectrogram image format. Combined with mel spectrogram data transformation techniques (i.e time and frequency masking [210]), they use a pipeline similar to many image instance discrimination methods to a learned representation that is language agnostic and is shown to transfer well to an emotion classification task, regardless of the spoken language.

A summary of the methods that learn an audio representation using Contrastive learning can be seen in Table 3.4.

3.4.4 Graphs

For relational and graph-structured data, contrastive learning has been successfully applied to learn both node, edge and graph-level representations.

The earliest approaches to learning representation from relational data that comes in the form of triplets *(subject, relation, object)* is Linear Relational Encoding (LRE) [208]. In this early work, the representation encoder is just a simple embedding layer for the *subjects* and *objects*, while the *relations* are represented as a matrix. The transform head in this case is a simple matrix-vector multiplication between the *relation* and *subject*, so that the resulting vector is closest to that of the *object*.

Later, Bordes et al. [24] introduced TransE, which learns a vector embedding for both the nodes and edges, and uses an additive transform head to represent relations as a translation in the embedding space. TransE uses an energy-based triplet loss to learn the embeddings and similar to LRE, the negative training pairs are created by corrupting the *object* node with random nodes from the data.

More recently, the Contrastively-trained Structured World Model (C-SWM) [149] uses a Graph Neural Network to model each state embedding as a set of objects and their relations. The base encoders consist of a CNN object extractor and an MLP object encoder, that turn an image into an abstract state representation.

Table 3.5: A summary of methods that applied contrastive methods on relational and graph-structured data. The color for defining similarity in query and keys encodes: Multi-sensory, Data transformation, Context-Instance, Sequential Coherence, Clustering. Colors for encoder represent: End-to-end, Online-Offline, Pre-trained. Colors for transform head represent: Projection, Contextualisation and Quantisation.

Method	**Query**	**Positive keys**	**Negative keys**	**Encoder**	**Transform head(s)**	**Loss**
LRE [208]	Concept + Relation	Paired Concept	Random concept	Embedding	Multiplication	Probabilistic loss
TransE [24]	Concept + Relation	Paired Concept	Random concept	Embedding	Addition	Triplet loss
Node2vec [101]	Query node	Neighbour node	Random node	Embedding	No	NEG
DGI [274]	Global graph	Local graph	Corrupted local graph	GCN	Readout average	Binary NCE
C-SWM [149]	State + Action	Next state	Random state	CNN and GCN	Addition	Pair loss
InfoGraph [262]	Global graph	Local graph	Random local graph	GIN	Readout summation	JSD MI estimator
Hassani and Ahmadi [107]	Global graph	Transformed local graph	Random local graph	Graph ConvNets	Readout + Pooling for global, non-linear for local graph	JSD MI estimator
GCC [220]	Sub-graph structure	Transformed sub-graph	Random sub-graph	Momentum GIN	No	InfoNCE

The graph Neural network heads then transform the state's representations and its corresponding actions (represented as one-hot vectors) into the state representation in the next time step. Similar to TransE, the state transitions between time steps is modeled as a translation in the embedding space and the entire world model is trained end-to-end with an energy-based hinge loss.

Focusing on learning useful node representations from general graphs, node2vec [101] aims to learn a node representation that is similar between neighbour nodes. The key contribution of node2vec is a family of biased random walk methods, allowing for a flexible notion of network neighbourhood (i.e positive keys). The model is trained similar to the Skip-gram model in word2vec, using negative sampling.

Veličković et al. [274] follows DIM [118] to propose Deep Graph Infomax (DGI) to learn node embedding by maximising mutual information between representations of local and global patches of a graph. The encoder is a Graph Convolutional network [151, 82] that summarises a patch of the graph centered around some nodes. A contextualisation head in the form of a *readout function* summarises the patch representations into a graph-level global representation so that all patches encode the most useful features present in the global features. The negative samples are patches from random graphs in a multi-graph setting or a corrupt function is used in a single-graph setting.

Also inspired by the mutual information maximisation between global and local structure of DIM, but with some design choices different from DGI [274], InfoGraph [262] focuses on learning graph-level representations. InfoGraph uses GIN [297] as the base encoder and uses sum over mean for the readout function, both of which are more suitable to learning representations at graph-level.

Combining both the multi-view and global-local mutual information maximisation objective, Hassani and Ahmadi [107] aims to learn both graph-level and patch-level representations for graphs. A graph diffusion is used to generate a different structural view of the graph, and then a sub-graph is sampled from both of the views. A dedicated GNN is used as the base encoder for each view, while the transform heads are shared between the two views. An MLP is used as projection head for the node representation, while a pooling layer followed by an MLP is used as the contextualisation head for the graph representation. A mutual information contrastive loss is then used to maximise the similarity between a local representation of one view to a global representation of another view.

Aiming to learn a structural representation of a graph without node attributes and labels, Graph Contrastive Coding (GCC) [220] simulates the augmentation-based instance discrimination task in computer vision. GCC treats each sub-graph as an instance and tries to learn a representation that captures similarity between sub-graphs by discriminating between these instances. A positive key is created by

applying a *graph sampling* transformation on that sub-graph. GIN [297] is used as the base encoder with a momentum encoder [110] for the keys and InfoNCE is used as the contrastive loss.

A summary of the methods that learn graph representations using Contrastive learning is shown in Table 3.5.

3.4.5 Multi-modal

The constraints enforced by the contrastive loss distance metric are not limited to embeddings from the same media modality. Contrastive learning has also been used to to learn cross-modal embeddings from two or more modalities that enhances the representation learned from a single data modality, especially for data that has limited labels.

In the most obvious way, the "views" from Contrastive Multiview Coding (CMC) [268] is straightforward to extend to multiple modalities. In this thesis, they experimented with views from L and ab channels from RGB colour images, or from one RGB frame and an optical flow feature at the same time.

The *Audio-Visual Correspondence* task is one example where it is desirable to have a joint representation space between representations extracted from the visual and audio modalities. The Audio-Visual Embedding Network (AVE-Net) [7] is an example where contrastive learning is applied to this problem. Two separate convolutional encoders for the vision and audio data streams are used. The audio which is 1 second in duration and is centered around the selected frame, is considered a positive pair, while negative pairs are extracted from different videos. This is different from the verification setting from previous work [6], where an MLP fusion network takes the concatenation of the two representations and outputs the final decision on whether the signals correspond. Instead, AVE-Net explicitly projects representations from each sub-network to a common embedding space through the use of a non-linear MLP head and measures correspondence through a contrastive loss using Euclidean distance in the embedding space. Since similarity between representations is explicitly enforced instead of implicitly learned in the fusion network as in [6], the embeddings learned by AVE-Net [7] are well-aligned and more suitable for cross-modal retrieval tasks.

Similarly, Cross-modal Audio Visual Instance Discrimination (Cross-AVID)[194] jointly learn the general representation from video using corresponding image frames and audio segments. In addition to contrasting between audio and visual representations of the same instance, they introduced a Cross-modal Agreement (CMA), a mining method that extends the set of positive pairs beyond just from a single instance. CMA measured the agreement of two videos based on both their visual and

Table 3.6: A summary of methods that applied contrastive methods on multimodal data. The color for defining similarity in query and keys encodes: Multi-sensory, Data transformation, Context-Instance, Sequential Coherence, Clustering. Colors for encoder represent: End-to-end, Online-Offline, Pre-trained. Colors for transform head represent: Projection, Contextualisation and Quantisation.

Method	**Query**	**Positive keys**	**Negative keys**	**Encoder**	**Transform head(s)**	**Loss**
CMC [268]	*L*-channel	*ab*-channel	Random *ab*-channel	Convolutional Net	None	InfoNCE
AVE-Net [7]	Query frame	Audio clip centered around query	Random audio clips	Convolutional Net	None	Euclidean distance + linear classifier
Cross-AVID [194]	Query video clip	Paired audio clip	Random audio clips	Convolutional Nets	MLP	Binary NCE
Patrick et al. [213]	Augmented query video clip	Augmented paired audio clip	Random audio clips	Convolutional Nets	MLP	InfoNCE
Afouras et al. [1]	Local features map of video	Global feature of aligned audio	Global feature of misaligned audio	Convolution Nets	Spatial Max-pooling	InfoNCE
Jiao et al. [134]	Query video frames	Aligned audio	Misaligned audio	Convolutional Nets	MLP	InfoNCE
Sun et al. [261]	Query video	Paired ASR text	Random ASR text	Video: 3D Convolutional Net + Transformer, Text: BERT	Transformer	InfoNCE
Ilharco et al. [126]	Object image	Paired text description	Random text description	Text: BERT, Image: Faster RCNN	LSTM + linear	InfoNCE
COALA [78]	Query audio	Paired tags	Random tags	Audio: ConvNet, Multi-hot tags: MLP	Non-linear	InfoNCE
CSTNet [146]	English speech	Paired translation text	Semi-hard mining translation text	Audio: ConvNet, Text: Word embedding + ConvNet	None	Triplet loss
CLIP [222]	Images	Paired captions	Random captions	Vision ConvNet or ViT, Text: Transformer	Linear projection	Symmetric cross-entropy

acoustic characteristics and if two videos have high agreement in both modalities, they are considered positive pairs.

Performing *within-modal* contrastive learning beyond the instance-level using the extended definition of positive pairs from CMA helps to improve the performance of Cross-AVID, and reduces the chances of the representation collapse phenomenon observed in cross-modal learning settings. Very similarly, Patrick et al. [213] performed visual audio cross-modal contrastive learning with a more principled approach to sampling and augmentation in an attempt to qualitatively measure the invariance and covariance, which they refer to as "distinctiveness", captured by the learned embedding.

Instead of contrasting cross-modal representations of different instances, Afouras et al. [1] used de-synchronisation to select negative samples by mis-aligning (shifting) the video and audio features. The global features from the audio signal for a frame is compared with the local features from the feature map of the vision network, resulting in an audio-visual attention map. A max-pooling layer acts as the context head to summarise the agreement between the audio and visual signals.

Jiao et al. [134] applied the misalignment objective to learn joint embeddings for ultrasound audio and the corresponding doctor's narrative speech. Applying contrastive learning in this setting is particularly helpful because this type of paired data is a lot easier to collect in a medical setting. Positive and negative pairs are defined based on spectrum of misalignment in time. Positive and "hard-positive" pairs are video frames and their corresponding or slightly misaligned audio clips. Negative and "hard-negative" are pairs of frames and audio clips that are even further misaligned from each other in time.

Instead of learning the correspondence directly between the visual and audio signals, in [261] video representations are learned by contrasting with representations from text captions extracted from an Automated Speech Recognition (ASR) system. The ASR sequences are encoded using a pre-trained BERT [61] model while a pre-trained S3D [294] model is used to extract visual features which are then fed into a shallow Transformer [272] network to construct a video-level visual embedding. The scoring function comprises another shallow transformer module that acts on the concatenated representations from the two modalities, followed by an MLP network that estimates the mutual information (MI) between the two inputs. The MI scores between them are again estimated through a softmax classification setting.

Not limited to jointly learning an embedding space, contrastive methods can also be used to learn a mapping between two separately-trained models of different modalities. Ilharco et al. [126] learned a probe to find the similarities between words and object images from a paired image captioning dataset. Even though the BERT [61] text encoder and the Faster RCNN [226] object detection model are trained

separately and not updated by the contrastive loss, the LSTM cells [120] and a linear project head can still map between words and object representations.

In the same spirit of learning representations from loosely aligned data, *COALA* [78] learns a shared embedding between audio and its tags, which are more readily available than a corresponding audio-transcript. In a different setting, Khurana, Laurent, and Glass [146] demonstrated a proof-of-concept approach to learn a translation network between English speech and its text translations in other languages. Their *CSTNet* used a triplet loss with a semi-hard negative mining method to learn both a cross-modal and cross-lingual representations.

Most notably is the success of Contrastive Language Image Pre-training (CLIP) [222], an approach to learn general a representation space from large-scale paired data of image and text captions. Instead of the generative objective of predicting the exact *next word* in the text caption, it employs a contrastive objective to match the meaning of the *entire* caption, resulting in a 4x efficiency boost in few-shot and zero-shot transfer tasks. The resulting fusion of the text and image representation space has enabled a plethora of downstream tasks such as image generation models from text prompts [227] [223].

Multi-modal learning, especially between text and images, is a fast growing topic. Contrastive methods continue to be the driving force behind many such methods [2] [296], due to their simplicity and scalability, and importantly, their expressive and versatile properties.

A summary of the methods that learn multi- or cross-modal representation using Contrastive learning is shown in Table 3.6.

3.4.6 Others

We conclude this section on applications of contrastive learning by looking at some others works that apply contrastive learning on other field such as reinforcement learning or that are different from the usual pre-train then transfer of contrastive representation learning framework in other modalities.

Not limited to learning representations, contrastive learning can also be applied to distill knowledge from a large pre-trained teacher network to a smaller student network, as demonstrated in Contrastive Representation Distillation (CRD) [269].

In addition to learning representations of observations in the environment, *CPC—Action* [102] is a variant of CPC that explored whether contrastive learning methods can also encode *belief states* (i.e its uncertainty) in its representation condition on the future action.

To improve the representation for reinforcement learning (RL) tasks, *CURL* [158] applied the instance discrimination task with a momentum encoder from MoCo [110]

to train model-free RL agents directly from the pixel observations. Due to the fact that many RL algorithms operate on a sequence of frames, the augmentations to create positive pairs are applied consistently across a consecutive frame stack as opposed to a single frame.

In an attempt to decouple representation and reinforcement learning, Stooke et al. [259] proposed the Augmented Temporal Contrast (ACT) for pre-training representations that are transferable to multiple RL tasks. Using the temporal consistency properties and a momentum encoder, augmented observations are contrasted with future observations in the same trajectory using the InfoNCE loss.

In a different vision application, Park et al. [211] proposed multi-layered patch-wise contrastive methods to enhance the performance of an unpaired image-to-image translation model. With the intuition that for a given patch of a style-transformed image, the corresponding patch at the same layer and spatial location should be more strongly associated with that patch than at any other patches at different spatial locations, InfoNCE contrastive loss is used to maximise the mutual information between patches at the same spatial location of both input and output images.

In other lines of work that try to learn representations in a greedy layer-wise manner instead of through an end-to-end approach using gradient descent, it has been shown that mutual information maximisation through the contrastive InfoNCE loss is particularly suitable for greedy optimisation. In this direction, Greedy InfoMax (GIM) [172] extends the approach of CPC [204] while Local Contrastive (LoCo) [295] improved the performance by extending SimCLR [41] with a modified overlapping architecture between local layers.

3.5 Discussion and Outlook

In this section we analyse and raise some questions about the current limitations and possible future directions for contrastive representation learning.

What kind of representations are learned by contrastive methods? Recent successes in transfer learning by instance discrimination contrastive pre-training [190, 110, 41] have raised the question of "what representation is learned from contrastive methods and why is it better than supervised pre-training" [312, 270]? However from the view of the Contrastive Representation Learning framework, the invariant and covariant features learned from the instance discrimination task are entirely decided by the augmentations techniques that create the positive pairs. To understand the effect of augmentations on the representation, one must take into account the bias of the dataset that it was applied to as well. As analysed in [219], models trained with an instance discrimination objective rely heavily on the oc-

clusion invariance property, which was induced by applying aggressive cropping on centred, single-object images from ImageNet [58]. Naively applying this "overfitted" set of augmentations on a different dataset with a more diverse composition of scenes can lead to unexpected behaviour in the representation. To successfully apply contrastive learning to other data sets and problems, one must be aware of the bias represented in the data together with the principle behind how positive and negative samples are produced (Section 3.2.2).

Contrastive loss needs more or no negative samples? Based on the theoretical guarantee of NCE and empirical evidence, the performance of contrastive learning methods benefit from comparison with multiple negative samples, which requires training on large GPU clusters and longer training times. One approach to alleviate this problem is to employ memory tricks such as the momentum encoder technique (Section 3.2.3) that can allow the incorporation of even more negative samples and is not limited to the batch size limited by hardware memory. Based on the assumption that negative samples are present just to prevent the representation from collapsing into one single cluster, another direction is to eliminate the need for negative samples altogether and impose additional constraints on the embedding space to prevent it from collapsing [100, 74].

Some methods follow the principle of redundancy-reduction, where the proposed novel loss functions not only impose constraints on the latent vectors between samples, but also between dimensions of the same latent vector as well [305, 16]. Many approaches inspired by the similarity between contrastive learning and self-distillation frameworks, such as [33] [207] [313], have explored approaches that still require data pairs but avoid explicit comparison between negative samples. Chen and He [44] explores different architectural designs to highlight the crucial components that enable a siamese network to learn without collapse of the representation. Preliminary results indicate that, as long as there is an asymmetry in the network architecture, even something as simple as a stop-gradient operation, it is possible to learn useful representations without requiring comparison with negative samples.

Beside quantities, qualities of negative samples are often neglected as sampled uniformly from the data distribution. More careful selection of negative samples has been shown to improve the convergence rate and performance of the learned embeddings on downstream tasks. This is consistent with hard negative and positive mining techniques, which has been a standard component in many metric learning applications.

This raises the question of a quality vs. quantity trade-off in employing negative samples for contrastive loss. Would it be possible to design a contrastive loss that employs both architectural constraints, perhaps for early stages of learning, and

uses hard negative samples to learn a more fine-grained representation in the latter stage?

What and how do different architectural designs affect the performance of contrastive methods? The separation between the transform heads and base encoder serves as a conceptual distinction to focus on transfer learning on downstream tasks, but in practice the distinction is not so clear cut. While the base encoders are mostly borrowed directly from supervised learning, with some modifications such as wider layers to capture more features, the best choices for projection and transform heads is unclear. In some cases the transform head is necessary (e.g to perform feature aggregation as shown in Section 3.2.2). Other possible choices are to not use any head, or to use a linear layer and non-linear multi-layers projection heads. In SimCLRv2 [43], empirical experiments show that the output of the second layer of a 3-layer MLP projection head is a better representation for transfer learning than the output of the base ResNet [108] encoder. In BYOL [100], in addition to the projection head from a high-dimensional representation embedding to a lower-dimensional metric embedding, a MLP "prediction network" projects metric embeddings of the online to that of the offline networks. This additional bridge between two embedding spaces is a crucial component for the success of the entire model.

These design choices are usually the result of empirical experiments specific to the architecture. The observations suggest a potential discrepancy in architectural design for supervised learning and representation pre-training, as well as potential for research in principles to design an efficient architecture for contrastive methods and representation learning in general.

Another under-explored topic is the specific form of the representation, which is currently treated as a simple vector for each input. With the ease of specifying invariant and covariant properties allowed by the contrastive framework, LooC [292] is an example where contrastive learning is used to concurrently learn multiple embedding sub-spaces, each of which is invariant to all but one transformation as specified by the distribution of positive pairs. Learning disentangled and compositional representations using contrastive learning is a promising research direction.

An asymmetric scoring function? Even though the learned similarity score has previously been used for retrieval and ranking applications, currently computing similarity or distance in contrastive learning is mostly used as a proxy task to learn representation. Can the learned similarity score be used in novel applications that were not previously possible?

An interesting possible extension for the scoring function is an asymmetric

one. The current literature on contrastive methods assume a simple symmetric distance/similarity relationship, but not all kinds of similarity are the same (for example the similarity between "dog-cat" should be different from "dog-animal"). Could a contrastive loss with non-transitive similarity relationships be developed?

Future of the contrastive loss function? As discussed in Section 3.2.5, the form of the contrastive loss is generally motivated from an energy-based margin loss, NCE-based classification or mutual information maximisation. The most popular form of contrastive loss belongs to the family of InfoNCE (and its variants such as NT-Xent), due to its efficiency and simplicity, with a well-grounded motivation from information theory. Can we design better contrastive loss functions that are more efficient in computation and memory, for example one that is more suitable to incorporate multiple positive keys for one query?

From which perspective can such a loss be developed? Even though contrastive losses motivated from mutual information have a strong body of theoretical support, as pointed out in [271], maximising mutual information alone can not explain for all the successes of contrastive learning methods.

One recent attempt to gain deeper insights into the inner working of these contrastive learning methods was by Balestriero and LeCun [15]. They showed that many of the proposed contrastive self-supervised methods, with or without negative samples, correspond to different spectral methods within the area of spectral manifold learning.

Looking at the contrastive loss from all the different perspectives may motivate the development of a new generation of contrastive losses.

Beyond learning representation with contrastive methods While this paper focuses on the majority of work that applied contrastive learning to learn representation, either supervised or self-supervised, the question of whether learning representation first is actually *necessary*, is still not settled. Even though there is ample evidence that representation learning on a general data stream benefits the performance of models when fine tuned on low-resource tasks, one can argue that if we know the task we want to be good at there are better ways to directly optimise for that task without explicitly dealing with the representation as a leaky abstraction [27]. Because contrastive learning only needs a definition of positive and negative distribution for pairs of samples, one can potentially define those just once for the entire data set or data stream, and optimise directly for a relevant task using the contrastive loss. Therefore contrastive methods can potentially extend task-based learning beyond the need for a static labelled dataset, as is the case for current supervised learning methods.

Through exploring contrastive learning approaches in this chapter, we have examined how pre-training of self-supervised methods can lead to broadly useful representations. Being able to learn such general and widely applicable representations for a wide variety of tasks, robust to different transformations, which are not limited by the need for hand-labelled data, are the foundation and inspiration for other representation learning approaches. Thus, we have directly addressed Research Question 1 on the architecture and training objective with the Contrastive Learning Framework. Similarly, we satisfactorily answered Research Question 2 on the principles and inductive biases for representation learning with our taxonomies of its components.

In the following sections, we will shift our focus to another representation learning problem, the challenge of learning higher-level semantics and abstractions, known as object-centric representation learning.

Chapter 4

Object-centric Representation Learning

Carving nature at its joints

Plato

In Chapter 3 we presented a deep dive into the topic of contrastive representation learning and covered many different approaches to building a robust and scalable representation. While the progress over the past few years has been remarkable, most of those progressions have been in the aspect of perception, like recognition, detection and segmentation.

Continuing our discourse from Section 2.5, in this chapter, we shift our focus towards the third research question in this thesis:"How to learn a hierarchical representation that captures increasingly complex and abstract concepts of the world?" Our approach will be through the lens of object-centric representation learning. This approach focuses on breaking down complex scenes into a structured set of interpretable and reusable elements, a crucial step in many aspects of artificial intelligence such as robotics manipulation and visual reasoning.

In this chapter, we first dive into the motivation and general goal of object-centric learning via the history and development of deep learning and AI in Section 4.1 and 4.2. We then define the problem setting with a concrete example for a proxy task in learning object-centric representation, as well as commonly used evaluation metrics in Section 4.3. Then we provide a comprehensive review of the state-of-the-art methods that have been developed in the field of object-centric representation learning over the recent years. This includes an overview of the various datasets used for advancing the topic in and state-of-the-art slot representation for object-centric learning methods and techniques in Section 4.4. Then we conclude with the most important challenge of scaling up these methods to work on real-world datasets in

Section 4.5.

Building on this foundation, chapters 5, 6, and 7 will present a series of experiments that address specific aspects of this complex problem. Each chapter will address a unique challenge in the journey to achieving effective object-centric representation learning, providing a more in-depth understanding of this area of research.

4.1 Motivation

Consider the ultimate aspiration for the field of Machine Learning, and Artificial Intelligence more generally: to develop an intelligent agent that is capable of observing, reasoning and planning its interaction with the real world. This is an encompassing challenge and can be naturally decomposed into two distinct problems: perception and reasoning.

Perception serves as the foundational step in developing any intelligent systems. It refers to an agent's capacity to comprehend its intricate environment across various spatial scales and depths, encompassing multiple independent objects and their interactions. In essence, perception is the ability of an agent to interpret the world through its sensory inputs. Given that these inputs are typically noisy, the system must possess the capability to filter and process the information to construct a meaningful internal representation. Reasoning, in the context of artificial intelligence, refers to an agent's ability to make inferences and draw conclusions based on the available information. Consider the example of an autonomous vehicle: perception involves detecting other vehicles, pedestrians, cyclists, road signs and signals, while reasoning encompasses making decisions related to steering and acceleration to reach the desired destination quickly and safely.

The conceptual division between perception and reasoning also aligns with the cognitive models of System 1 and System 2 thinking in human cognitive processes, as expounded by Kahneman [139]. System 1 corresponds to fast, automatic, and subconscious cognitive processes. This mode of thinking is characterised by its speed and efficiency, making it suitable for tasks related to perception or quick decision-making. Therefore, System 1 thinking is susceptible to biases and cognitive shortcuts, including overlooking details, confirmation bias, and a tendency to ignore contrary evidence. System 2, on the other hand, operates at a slower pace and requires more effort. It is associated with logical processes and engages when the task at hand is more complex, necessitating deliberate and conscious effort for resolution. System 2 allows individuals to invest more time and energy in problem-solving, enabling a more thorough and analytical approach.

Historically, the development of AI has been dominated by two paradigms: symbolic AI such as expert systems motivated by the aspect of reasoning, and connec-

tionist approaches such as neural networks concerned with the challenge of perception. In the following sections, we will briefly review the approaches addressing the challenges associated with each paradigm and explore methodologies for bridging the gap between them. This analysis aims to provide insights into the evolution of AI methodologies and provide context around contributions that have motivated the development of object-centric representation learning methods.

4.1.1 From Perception to Reasoning

Symbolic Methods for Reasoning Symbolic AI, or *classical* AI, dominated the field from the 1960s to the 1990s, placing logic and reasoning at the core of intelligence. Systems based on symbolic reasoning rely on rules, logic, and explicit representations of knowledge. This approach involves injecting human knowledge into computer systems through human-readable symbols concerning *objects* and their *relations*, utilising logic to create rule-based systems for manipulating such symbols [199]. In Natural Language Processing (NLP), symbolic AI systems leverage rules, lexicons, and grammar trees as knowledge symbols for language understanding. Examples of symbolic AI applications include expert systems, knowledge-based systems, the semantic web, and automated theorem provers.

Symbolic AI offers interpretability and trustworthiness by providing logical conclusions using explicit rules and facts. Moreover, by design, it systematically generalises and infers new knowledge through logical inferences on its existing knowledge base. However, its reliance on handcrafted, explicit knowledge and rules poses serious limitations. In certain domains, the cost of collecting and building such knowledge-based symbols can be prohibitive, constraining development and usage. In other domains, constructing such symbols may be challenging or impossible. While effective in well-defined domains with explicit rules and relationships, symbolic AI struggles with handling uncertainty and learning from extensive datasets.

Connectionist Methods for Perception In contrast to the top-down dictation of knowledge and logic in symbolic AI, connectionist methods adopt a bottom-up approach in building intelligent systems, employing statistical and learning-based methods such as Deep Learning. Neural networks, particularly deep learning models, excel at learning patterns and representations from data, proving highly effective in tasks such as image recognition, natural language processing, and speech recognition. However, they may lack transparency and interpretability.

For a more detailed overview of Machine Learning and Deep Learning, refer to Chapter 2 earlier.

Despite the substantial success of Deep Learning in the past decade, its impact

remains confined mainly to perception tasks, including classification, detection, and segmentation though recently we have seen the emergence of strong interest in generative forms of AI. Even in the domain of language modelling where LLMs' successes entered the public splotlight, it is still prone to hallucination and fails to generalise to basic relations not present in the training data.

The application of deep neural networks to tasks requiring higher-level cognition, such as visual and textual reasoning, is still limited and demands web-scale datasets and supercomputing-scale compute resources. Additionally, interpreting the inner workings of a trained neural network is challenging. The final output of a deep network is obtained through successive layers of non-linear transformations, making neural networks often regarded as "black boxes" to human observers, owing to both the scale and nature of computation involved.

As highlighted earlier in Section 2.5.4, the integration of learning and reasoning capabilities remains a central challenge in the broader field of Artificial Intelligence research.

Hybrid Neural-symbolic methods An insightful observation at this point is that the strengths and limitations of symbolic AI and deep learning complement each other. While symbolic AI relies on handcrafted knowledge, deep learning excels at learning useful representations of inputs. Despite the data-hungry nature of deep learning and its limited generalisation beyond training distributions, symbolic AI offers a systematic template for generalisation through logical rules and inference based on objects and their relations.

The synergy between symbolic reasoning and neural networks, aiming to leverage the strengths of each approach, is the primary focus of the subfield Neural-Symbolic AI, also known as the 'hybrid' architecture [80].

Various approaches exist for building such hybrid systems [55]. Researchers have explored incorporating symbolic reasoning components into neural network architectures. For example, Neuro-Symbolic systems introduce symbolic structures like graphs or logic rules to guide neural network learning. Another approach involves embedding symbolic knowledge into neural network models, either by training neural networks with structured symbolic inputs or using embeddings to represent symbolic entities.

The common challenge for all methods in this line of research arises in the interaction between neural network and symbolic components. Optimising deep neural networks typically relies on an end-to-end differentiable architecture, which classical symbolic components inherently lack. Additionally, the distributed representations of neural networks may be sensitive to small disturbances in input and may not provide a stable symbol-like knowledge characteristics of classical AI approaches.

Achieving seamless interaction between symbolic and neural components remains a challenge. Integrating different representations and reasoning paradigms requires addressing issues of interoperability and communication between modules.

Structured Representation and Modular Neural Networks Another line of research acknowledges the importance of symbol-like manipulation and reasoning in classical approaches but aims to tackle it within the connectionist framework. The core motivation for this approach is the recognition that learning is the only approach that scales up with data and compute resources. The ability to generalise systematically and to perform reasoning in a symbol-like manner is a by-product of more efficient learning. The philosophy of this approach is often referred to as: neurons all the way down and learning all the way up, highlighting the lack of any handcrafted explicit symbolic structures.

In the field of deep learning, this line of thought is still in its infancy and is often advocated under different names. LeCun [159] advocates for a modular system based on deep neural networks in which latent vectors represent knowledge symbols and performing arithmetic on them corresponds to reasoning. Similarly, Bengio [20] advocates for incorporating more "consciousness priors" into traditional deep neural network architectures to tackle high-level reasoning and planning tasks. Battaglia et al. [18] explores the use of *relation inductive biases* in the form of graph networks to facilitate learning about relations, entities and the rules for composing them. A more recent survey by Pfeiffer et al. [214] reviews several threads of research in a more modular deep learning architecture. Veličković and Blundell [273] follows a slightly different direction and proposes to replicate classical computer algorithms with the machinery of neural networks as a form of reasoning.

A fundamental challenge in progressing this line of research lies in the ability to learn structured representations that mimic symbols in classical AI, without relying on handcrafted knowledge. In the following section, we will pinpoint a characteristic of the representation learned by the current generation of neural networks that hinders their capacity to acquire such structured representations.

4.1.2 Entanglement of Semantics in Representations

Currently, the focus of learning visual representations primarily revolves around large scale datasets like ImageNet [58]. While these datasets are diverse when aggregated, they often carry a strong yet subtle assumption: a single dominant object in each image.

Taking the example of AlexNet [157], a pioneering deep learning model that initiated the current wave of Deep Learning research not only by demonstrating

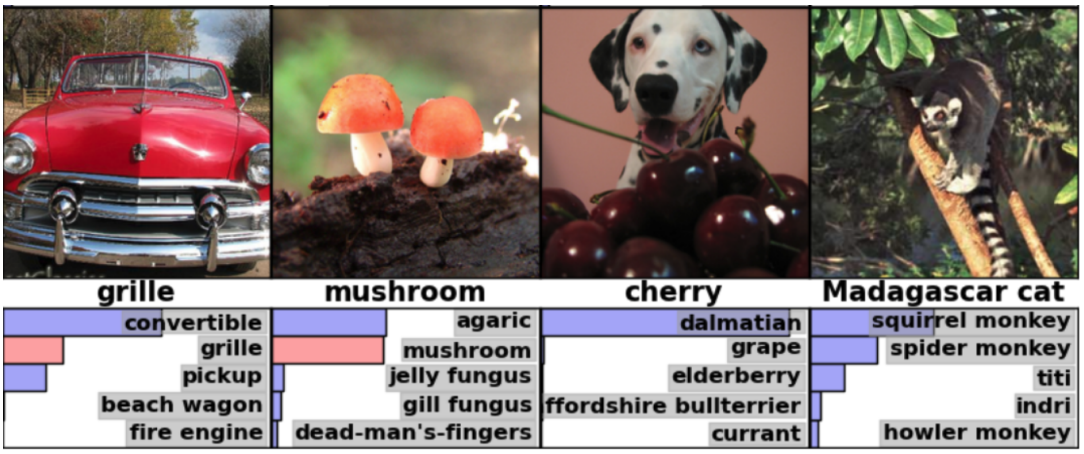

Figure 4.1: Figure produced by Krizhevsky, Sutskever, and Hinton [157] showing some test images from ILSVRC-2010 dataset [58], including their true labels and probabilities for the top 5 classes predicted by AlexNet.

a remarkable leap in classification performance at the time, but arguably by also providing valuable insights through its failure cases. In Figure 4.1 taken from the original paper showcasing failure cases ([157]), shows that even when the model's predictions were incorrect, the errors often made sense to humans. These errors mirrored common human mistakes, such as uncertainty between "mushroom" and "agaric", or confusion between a "Madagascar cat" and a "squirrel monkey". These are examples of mistakes due to fine-grained visually and semantically related concepts. In the first example, the model predicted "convertible" car which includes the human label as "grille", which is a misprediction from an ambiguous part-whole relationship.

Let our focus turn to the third example in Figure 4.1, where the model predicted the "Dalmatian" dog breed, followed by "grape, elderberry," and so on. Looking closely, we find that, unlike most images that prominently feature a distinct dominant object in the foreground, this is an example where there are more than one prominent objects in a scene. In these ambiguous scenarios, multiple interpretations could be considered correct. This ambiguity leads to the entanglement of information in the representation space, and the consequence of this entanglement becomes evident in misclassifications, as observed in the prediction of the "Dalmatian" breed in conjunction with references to fruits like "grape" and "elderberry." The model's representation encodes features from both the dog and the surrounding cherry, resulting in a blending of semantics in its representation.

In addition to misclassifications, this entangling of semantics in the representation space is also a prevalent source of error in deep neural networks, introducing spurious correlations in predictions [81, 193]. One promising approach is object-centric representation learning, which seeks to learn representations that align with the

causal mechanisms governing our physical world. It do so via learning to produce a set of stable representation that factors the complex visual scene into their high-level objects. Such representations are thought to be more robust to out-of-distribution data and support more complex tasks like reasoning and control. In addition to errors on perception tasks by directly making predictions from such entangled representations, it could also pose numerous challenges when these scene-centric vector representations are employed as building blocks for other neural network modules for planing and reasoning. Unravelling this complexity is a key motivator to advance object-centric representation learning, aiming to discover the independent constituents, and hence disentangling and refining the information encoded in the representations to enhance the accuracy and robustness of artificial intelligence systems.

4.2 What is Object-centric Representation Learning ?

Object-centric representation learning stands out as one of the most promising approaches within deep learning for acquiring structured representations, especially of visual data. Rooted in the connectionist paradigm, it is firmly grounded in data and learning methods while striving to achieve a structured representation that exhibits symbol-like qualities, making it more amenable to manipulation and suitable for downstream applications.

4.2.1 Goals

Object-centric representation learning, as implied by its name, endeavours to learn representations of individual objects within a scene rather than acquiring a representation of the entire image or scene [73, 170]. The motivation behind this approach is drawn from the intuitive notion of objects in human cognition, aiming to capture and leverage the way humans naturally perceive and interact with their surroundings [253]. To do so, it must be able to perceive and factor complex and unstructured visual inputs into their constituent objects, representing them independently.

Such representations have the potential to radically transform and simultaneously address many current challenges in learning dense scene-level visual representations such as:

- Sample efficiency: Object-centric representation learning stands out as a potential catalyst for enhancing sample efficiency across a spectrum of downstream tasks. By focusing on capturing essential features related to individual

objects within a scene, the learning system can distil pertinent information more effectively, reducing the need for extensive datasets to achieve optimal performance.

- Structural understanding and abstraction: Having stable building blocks is a prerequisite for forming structural understanding and building layers of abstractions. By dissecting scenes into discernible objects and their interrelationships, the learning system gains the ability to abstract essential structural elements, enabling a more sophisticated comprehension of intricate visual contexts.

- Systematic generalisation: As a consequence of having interchangeable and independent representations, and potentially a better structural understanding and abstraction, the learning system becomes adept at systematic generalisation, extrapolating knowledge to novel scenarios with increased accuracy. Aligning a neural network's internal representation to a higher level of abstraction more similar to that of human could also reduce mistakes from spurious correlations, shortcut learning and surface-level statistics.

- Reasoning: As the learning system becomes attuned to the hierarchical and relational aspects of objects, it lays the foundation for more advanced counterfactual or causal reasoning capabilities. This, in turn, facilitates a deeper understanding of complex scenarios, empowering the model to make informed decisions and predictions in diverse and more complicated situations.

4.2.2 The Binding Problem

The entanglement of semantics in neural network distributed representation spaces, as discussed earlier, is a characteristic intrinsic to these systems. Greff, van Steenkiste, and Schmidhuber [96] defines the root cause of this behaviour as the *binding problem*: "The inability of existing neural networks to dynamically and flexibly *bind* information that is distributed throughout the network. The binding problem affects their ability to form meaningful entities from unstructured sensory inputs (segregation), to maintain this separation of information at a representational level (representation), and to use these entities to construct new inferences, predictions, and behaviours (composition)."

The binding problem has its roots in neuroscience, and is used to explain information processing in the brain, including sensory and cross-sensory binding (e.g., colour, shape, texture, voice), binding across time with motion, and binding with actions or semantic knowledge and memory.

In current neural networks, the information routing process is largely determined by predefined architectures and fixed parameters post-training. There is limited dynamic capability to segregate and group information, mostly occurring at the level of image patches or word tokens. Addressing the binding problem in the connectionist approach involves discovering the right inductive biases that enable the emergence of symbol-like representations through learning.

Greff, van Steenkiste, and Schmidhuber [96] further dissects the binding problem into three subproblems:

The Segregation Problem: Involves turning unstructured and complex input data into meaningful entities, i.e., objects. The concept of an object is inherently vague and context-dependent, requiring continuous and dynamic factorisation of the input stream. The segregation problem is somewhat analogous to classical computer vision tasks, such as object detection and segmentation. However, the goal is not merely obtaining predictions for location or semantic classes but addressing the challenge of dynamically segregating input data.

The Representation Problem: Focuses on binding, maintaining and representing the segregated information into independent entities, so called "object-representation". These object representations should behave like symbols in classical AI, serving as building blocks for downstream neural processing modules. They should be self-contained, separating relevant information, yet capable of being grouped and assembled into more complex structures.

The Composition Problem: Addresses how these representations can interact, exchange information, and be composed into useful, novel representations that can generalise systematically for inference, prediction, reasoning, and planning. The output of the composition step could potentially inform the segregation step in a top-down fashion.

These three challenges from segregation, representation, and composition within the binding problems present numerous research opportunities and problems. The topic of object-centric representation learning currently primarily focuses the segregation and representation problem, with the goal of discovering and representing objects in a visual scene, discussed in more detail in the next section.

4.3 Learning and Evaluation

Object-centric Representation Learning (Object-centric Representation Learning (OCRL)) methods operate within the domain of multi-object images or video datasets,

contrasting with commonly used 'object-centric' datasets like ImageNet, which primarily contain a single dominant object in the foreground. The definition of an object is not explicitly constrained and depends on the specific dataset. It could encompass simple geometric 2D or 3D shapes for straightforward datasets or extend to everyday household items, vehicles, or human entities in more complex datasets.

In the context of visually complex scenes, whether in images or videos, comprised of multiple individual objects, the objective is to automatically discover these independent components. The goal is to parse and bind the information of each object into independent representations. Each object representation should adhere to a common format, be interchangeable, and together, they should collectively describe the original input data.

4.3.1 Task: Unsupervised Object Discovery

In pursuit of advancing Object-centric Representation Learning (OCRL), our focus centres on the unsupervised Object Discovery task. As previously highlighted, the notion of an object is ambiguous, and the set of all possible objects is infinite. For meaningful progress toward practical and useful object-centric learning, these approaches must not rely on human supervision, but rather be unsupervised, self-supervised, or semi-supervised through additional contextual signals.

The pretext task used throughout this thesis and commonly found in the literature is instance segmentation. Instance segmentation is a computer vision task that involves identifying and separating individual objects within an image. It includes detecting boundaries and predicting the exact pixel-wise mask of each individual object instance in an image, assigning a unique label to each object. This task stands as a special form of image segmentation that deals with detecting instances of objects and demarcating their boundaries. It provides more detailed and sophisticated output than conventional object detection algorithms. Unlike semantic segmentation, instance segmentation also differentiates between different objects belonging to the same categories.

Contrary to normal instance segmentation models, in object-centric learning, the ultimate goal is not to learn to predict the location of object instances from an entire scene representation. Ideally, object-centric learning methods would bind all the relevant information of an object into its corresponding representation. Each independent representation would then be used to extract the information it contains for each object, in this case in the format of segmentation masks.

By utilising traditional instance segmentation as our pretext task and constraining ourselves to the unsupervised setting in addition to generate segmentation masks from the individual representations, this approach can be used to make progress to-

ward our objective of object-centric learning.

4.3.2 Metrics

Independently from the task of training of supervised segmentation, in the unsupervised settings we need different metrics for both learning and optmisation to more accurately reflect the performance of techniques.

Reconstruction Error: MSE Since most Object-centric learning methods, as will be discussed in 4.4, are based on an autoencoding framework, models are optimised with a reconstruction metric between the ground truth and final predictions from all object representations. A common reconstruction metric is Mean Squared Error (MSE), defined on the ground truth $\mathbf{x}$ and the prediction $\hat{\mathbf{x}}$ as:

$$\text{MSE}(\mathbf{x}, \hat{\mathbf{x}}) = ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 = \frac{1}{D}\sum_{i=1}^{D}(x_i - \hat{x}_i)^2$$

where D is the dimensionality of all predicted locations, i.e all pixel locations and colour channels for an RGB image. It can also be seen as minimising the log-likelihood of the data distribution and a Gaussian distribution of each input points.

This is a direct and straightforward metric to measure and optimise for during the training process. Our premise is that given appropriate architectural bias, a model with lower reconstruction error should directly translate to a model that can perform better at discovering objects.

Adjusted Rand Index (ARI): To directly evaluate the segregation capability of object-centric methods, we need to measure its ability to segment an image or video input using its slot representation. In the unsupervised object discovery setting, since an object can potentially bind to any instance slots, there is no clear correspondence between the ordering of objects in the slots and the ground truths, thus making it unsuitable for more traditional segmentation metrics like mIoU. Adjusted Rand Index (ARI) [124] is a clustering similarity metric that is invariant to permutation in the ordering of clusters. This has made ARI a standard metric used to evaluate unsupervised object-centric segmentation in prior works [28, 138, 150].

We now describe in detail how to compute ARI. We have a set $S = s_1, s_2, \ldots, s_n$ of n elements and two different ways to partitioning this set $X = X_1, \ldots, X_r$ and $Y = Y_1, \ldots, Y_s$ with r and s subsets respectively. Given a pair of elements s_i and s_j, assign a label 1 if they belong to the same cluster in the first clustering Y or 0 otherwise. Now consider the binary classification task to predict whether a pair of elements belong to the same cluster in the partitioning X. In this binary

classification task:

- True Positive (TP) is the number of pairs of elements that *belong* to the same subset in X, and belong to the same subset in Y.
- True Negative (TN) is the number of pairs of elements that do not belong to the same subset in X and do not belong to the same subset in Y.
- False Positive (FP) is the number of pairs of elements that belong to the same subset in X but do not belong to the same subset in Y.
- False Negative (FN) is the number of pairs of elements that do not belong to the same subset in X but belong to the same subset in Y.

The Rand Index (RI) can be thought of as the accuracy for this classification task:

$$RI(X, Y) = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{\binom{n}{2}}$$

where the denominator equals the number of pairs from n elements.

The Rand Index ranges from 0 to 1, with 0 meaning complete disagreement for any pairs of elements between two clusterings while 1 means the two clusterings are identical, up to a permutation of the partitions. This is not ideal since it does not take into account clustering by chance.

The Adjusted Rand Index (ARI) adjusts to have a value of 0 for the expected number of agreements with a random baseline. Let $x_i = |X_i|, y_j = |Y_j|$ be the number of elements in their corresponding subsets and $n_{i,j} = |X_i \bowtie Y_j|$ is the number of elements that are in both subsets X_i and Y_j, then the expected RI by random chance is:

$$\text{Expected RI} = \frac{\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}}{\binom{n}{2}}$$

The Adjusted Rand Index is then computed as:

$$ARI(X, Y) = \frac{\sum_{i,j} \binom{n_{i,j}}{c} - \text{Expected RI}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \text{Expected RI}}$$

To evaluate the accuracy of the segmentation task, we consider the set of all pixels in a frame and the segmentation indices as the clustering assignment. To take into account consistent of object identity over time, we simply consider the set of all pixels from all frames simultaneously. The video ARI metric will then measure both object segmentation and tracking over time. Since object-centric methods tend to over-segment objects into the background category, following prior work [28] only the ground truth foreground classes are used for evaluation. This metric is then referred to as ARI-FG.

4.4 Slots Representation

In this section, we will provide an overview of "slot representations", the most popular format for learning object-centric representations. Then we categorise the methods based on the different way each slot is associated with objects in the visual scene following Greff, van Steenkiste, and Schmidhuber [96], and provide a brief overview of the methods in each category in their chronological order of development.

In traditional visual representation learning, a scene is typically represented by a single vector summarising its global information. To extend this scene-level representation to an object-centric representation, the most straightforward method is to equip each object with its own representational vectors. In the object-centric literature, these are often referred to as "slots" to denote their ability to dynamically bind and represent different information depending on the input data without being hardcoded.

Object-centric representation learning methods learn to segregate and bind each slot to the information of one object, and collectively, all slots will represent the entire visual scene. As each slot being represented has its own vector, each object representation is independent by nature. A change in one object would only affect one representation in a slot, whereas modifying or deleting a slot would correspond to a local change in the visual scene for that object. By the nature of parameter sharing, all slots are embedded in the same vector space, where each object representation will have a common format.

Together, slot representations provide a way to *holistically* encode a visual scene via its *independent* objects with a *common format* while retaining the powerful properties of *distributed representations* of deep neural networks.

By far, slot representations have been the most popular and successful approach in the field of object-centric representation learning. In this section, we will cover the development of object-centric representation learning methods over the past few years. We will do so by looking at the different inductive biases used in segregating objects' information into slots, including: categorical slots, sequential slots, spatial slots and instance slots.

4.4.1 Category Slots

One of the earliest approaches to routing information of different objects to different slots is by transforming the representation based on the object category [115]. These methods leverage semantic information to segregate and represent distinct categories of objects. Work in the area of Capsule Networks [116, 233] is the most representative line of work for this approach. Inspired by the goal of capturing part-whole relationships [114], each capsule in the network is learned to capture a specific

kind of object or a part of it.

However, this approach comes with certain limitations. By assigning a fixed representational capacity to different object categories, it becomes computationally expensive and wasteful, especially when not all object categories are present in the input. Additionally, by separating slots by their semantic category, it restricts the ability to separate and represent multiple objects of the same category simultaneously.

4.4.2 Sequential Slots

Methods in this category typically impose an order on the slot representations and the routing of information. This mimics how humans and animals use eye gaze to direct their attention to different aspects of a visual scene. Object representations are sequentially predicted one after another, often using recurrent mechanisms such as Recurrent Neural Networks.

A foundational method in this category is Attend, Infer, Repeat (AIR) by Eslami et al. [75]. This approach proposes the use of a Recurrent Neural Network to iteratively attend to and perform inference for one object at a time. What sets this approach apart from others is its capability to learn the use of a variable number of slots, depending on the input.

Sequential Attend, Infer, Repeat (SQAIR) [156] extends the recurrent aspect of AIR from objects in still images to objects in video. It introduces a new *propagation* phase responsible for updating and forgetting object slots from the previous timestep based on new observations, carried out in a recurrent manner.

However, due to their recurrent nature, performing inference to obtain representations for all slots can be computationally expensive. This is particularly costly during training, where the autoencoding step for inference cannot be parallelised over objects (for AIR) or over time steps (for SQAIR).

Another approach, MONet (Mult-object Network) [28], also follows a sequential approach but only applies it to the information routing step. MONet implements an attention network that recurrently outputs a soft object mask at each step from the input image and scope mask of what pixels have not been fully accounted for so far. Subsequent steps take the remaining unexplained scope to predict another object mask. Starting from a full scope of unexplained pixels, this process is repeated for the first $K-1$ slots, with the last slot K taking the remaining unexplained scope from the previous steps. By sequentially obtaining attention masks, MONet can parallelise the inference step by masking the input image with the masks and encoding and decoding them in parallel. This helps somewhat alleviate the disadvantage of sequential processing of object slots at the cost of higher compute and

memory requirements in the autoencoding step.

GENESIS [73] extends MONet by introducing an autoregressive model between slots to enable the application of novel scene generation.

4.4.3 Spatial Slots

Spatial slots refer to methods that bind object information into slots based on their location in the frame. This concept is akin to feature maps in Convolutional Neural Networks (CNNs) or patch tokens in more recent Vision Transformer architectures, which are popular for object detection and segmentation tasks [31]. Some earlier works [236, 255] that use these feature maps are also relevant for reasoning tasks, with a focus on learning about the relations among feature vectors in a feature map in a spirit similar to object-centric learning.

More relevant are methods that learned spatial feature maps with more explicit object structures. SPAIR [51] extended the AIR framework with spatially invariant object-like features such as "what," "where," and "depth." SPACE [165] further augments spatial features with features indicating the presence of objects at each spatial location. SCALOR [133] directly uses CNN's feature maps to generate object proposal maps before recurrently updating the object slots based on the SQAIR framework.

SIMONe [138] is a notable work in this category that extends learning both object and frame representations from videos. A multi-object input video is first encoded into spatiotemporal features, and the 3D feature map is jointly processed with a Transformer. The final object slots are obtained by summing all the representations at each spatial location over time, while the frame representation is obtained by summing all the spatial representations at each timestep. This approach exhibits remarkable compositional properties, enabling the generation of videos composed of objects from one video with the camera trajectory of another by using object and frame latents encoded from different videos.

Routing information based on spatial location enables efficient parallel processing and provides a strong inductive bias that aids object discovery. However, it comes with certain disadvantages. Since objects are tied to their locations in the image, the number of slots is tied to the grid resolution. This approach is also sensitive to the size of objects, where a large object can be represented by multiple slots simultaneously, while small objects may compete to be represented by a single slot, leading to the entanglement problem described earlier but in a different scale. Similar to category slots, spatial slots can be wasteful when there are few objects in a scene, and the network dedicates significant capacity to representing simple background slots.

4.4.4 Instance Slots

The most general form of slot representations is to bind object information for each instance in a scene. Unlike category slots, multiple instances of the same category can be represented in different slots. Unlike spatial slots, there is no direct correspondence between object location and its slot representation and unlike sequential slots, each instance is treated as independent and can be efficiently processed in parallel.

Earlier approaches [99, 94] treat the problem of detecting instances as a form of perceptual grouping and clustering. Neural Expectation-Maximization (N-EM) [95] implements a differentiable clustering method that simultaneously learns to group and represent individual clusters. IODINE is an iterative method over VAE [148] like MONet [28], but instead of iteratively sequencing over slots, IODINE binds objects to slots in parallel but iteratively refines these over time. Improved upon the prior version, GENESISv2 [72] uses a differentiable clustering process on the pixel embedding to infer and learn object representations without the need to specify the number of slots as a hyperparameter.

Slot Attention [170] is a notable approach that embraces the success of attention mechanisms in deep learning to perform object grouping in parallel with the expressive attention mechanism. Slot Attention is a type of attention mechanism that encourages slots to compete to explain each input position. In Slot Attention, the slots acts as the query and the visual features map are the keys and query. Unlike normal attention, Slot Attention applies softmax normalisation over the query dimension, letting the slots compete with each other to explain each position in the feature map. This creates a form of parametric clustering algorithm with the centroids being the slot representations. Due to its performance and efficiency, Slot Attention has quickly become one of the most popular methods in the field of object-centric representation learning. A core difference of slot attention and normal attention is the axis over which the softmax operation was applied. Due to its performance and efficiency, Slot Attention has quickly become the most popular method in the field of object-centric representation learning.

4.5 Scaling to Visually Complex and Real-world Data

The subfield of object-centric representation learning is still relatively young and is experiencing rapid development. Earlier methods explored various approaches for segregating and representing object information to varying degrees of success. These early experiments are trained and evaluated on simple synthetic datasets and

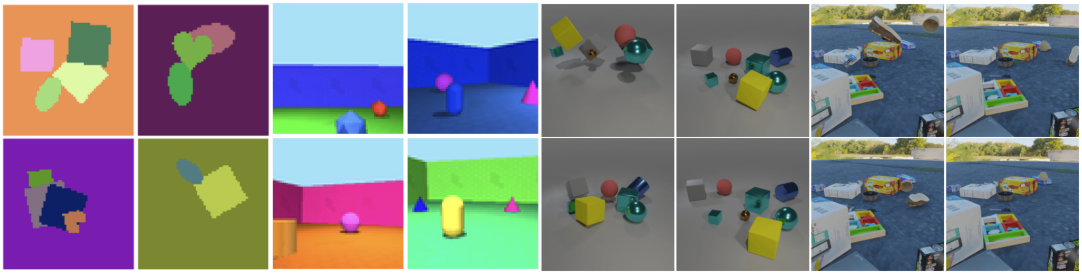

Figure 4.2: Representative datasets for multi-object datasets used for study object-centric representation learning over time, increasing in visual complexity. From left to right: Multi-dSprites, Object Rooms, CLEVR, MOVi-E datasets.

mainly aims to validate the central hypothesis on the feasibility of object-centric representation. The challenges faced by current object-centric learning approaches underscore the need for innovative solutions to achieve scalability and handle more complex scenes.

4.5.1 Datasets

Over time, increasingly more complex synthetic datasets were designed to test various aspects of object-centric systems, furthering progress in this space. The benefit of synthetic datasets lies in having multiple paired ground truth annotations, such as depth, flow, and surface normal, which are expensive to collect for real-world datasets.

Earlier works, such as [99, 94], used very simple 2D datasets like shapes [224], where images contained a few randomly placed geometric shapes, sometimes with overlap. Another variant, Multi-MNIST, was generated using digits from the MNIST dataset [162] as more complex objects. These datasets aimed to reduce visual complexity to a minimum to evaluate methods on their ability to bind shapes under varying translations, rotations, and overlap conditions.

Later methods increased the visual complexity by adapting the dSprites [181] dataset, which comprises 1 to 4 randomly chosen sprites placed onto a single image with a uniform randomly coloured background. Object Rooms and its video variant [28] maintain the same complexity but render scenes in a 3D environment.

The CLEVR dataset [136] features realistically rendered multi-object scenes with simple 3D objects on plain coloured backgrounds, providing a major advancement for object-centric learning methods to work with visually complex input. However, despite its realistic appearance, CLEVR objects only contain uniform colours with a clean background. To address this limitation and further challenge object-centric methods, CLEVRTex [142] augments CLEVR with more varied texture colours, introducing challenging foreground and background separation.

In the temporal domain, CATER [83] builds on CLEVR to generate videos with

moving objects and large camera movements, extending the challenges to object tracking under occlusions and over longer durations.

More recently, the MOVi dataset [97] was introduced, and comprises multi-object video datasets with increasing complexity across its 5 variants. MOVi-A follows the CLEVR setup, rendering basic geometric shapes on a plain background. MOVi-B incrementally increases difficulty with more complex objects, holes, and diverse backgrounds. MOVi-C represents a significant quality leap, utilising scanned 3D everyday objects (e.g., mugs, shoes) on photographed background images. MOVi-D raises the level of difficulty further by increasing the number of objects per scene (10 to 20 static objects and 1 to 3 moving objects). Finally, the MOVi-E variant introduces moving camera motion on top of MOVi-D, producing the most challenging dataset of all. This variant contains many small, real-world scanned objects — both static and moving — that interact with each other, all while being captured via a moving camera. The addition of moving camera motion adds another layer of complexity to the dataset, making it a valuable resource for evaluating object-centric learning methods in real-world scenarios with dynamic scenes and changing perspectives. Each variant consists of 10,000 videos with 24 frames each, rendered at 256x256 resolution. The MOVi dataset provides a comprehensive evaluation platform for various aspects of object-centric learning methods, covering both spatial and temporal challenges.

In Figure 4.2, we display a few representative samples of the datasets used in the field of object-centric representation learning overtime, highlighting the increas in visual complexity of the input domain.

4.5.2 Slot Attention for Video

The most recent methods aiming at scaling up object-centric methods are all built on top of Slot Attention, as discussed earlier. Among those approaches, Slot Attention for Video (SAVi) [150] provides a comprehensive framework for building object-centric learning methods, in particular on videos.

In SAVi, slot representations can be initialised from the object's bounding boxes of the first frame. This provides a weak supervised signal for the object discovery task, and it is especially helpful for visually challenging datasets like MOVi-C. This step can also enable interactive applications, allowing humans in the loop to resolve questions about relevant objects. From the weakly supervised initialisation, the object representations are recurrently updated over video frames, with the slots being naturally carried over from the previous frame as the initialisation for the next frame.

In each frame, the slot representations are updated via two steps: correction

and prediction. The correction step implements the Slot Attention mechanism to update the slots via the feature maps from the visual encoder of the current frame. Optionally, this correction step can be repeated many times to further refine the information routing of objects to slots. After the slots are "corrected" by the visual information of the current timestep, the prediction step further allows object slots to interact and exchange information between each other via a shallow Transformer network. The aim of this step is to enable the slots to model their interactions in the current frame before proceeding to the next.

At each step, slots are independently decoded to the target signals before being combined together. The reconstruction error between the combined prediction and the target provides the training signal for the entire network. SAVi further uses extra paired data such as optical flow to help guide the object discovery process based on object motion.

In contrast to SAVi, SAVi++ [71] applies scaling techniques in the architecture design, drawing inspiration from the broader deep learning literature. The method combines these techniques with additional geometric signals such as depth to scale up an object-centric learning method to a real self-driving dataset [263] for the first time. This indicates a significant step toward applying object-centric learning in real-world complex scenarios such as autonomous driving.

In summary, the SAVi framework provides a versatile and useful foundation for end-to-end learning in object-centric methods. The following chapters will explore specific sets of experiments targeting different components in this pipeline. Chapter 5 will address the challenge of representation dynamics with a focus on a novel discrete object-centric representation. Chapter 6 will concentrate on the object decoder components, exploring methods to enhance efficiency while retaining the desired properties of slots. Finally, in Chapter 7, the aim is to scale up object-centric methods by incorporating a pre-trained vision model as an additional signal, without relying on supervised data such as optical flow or depth information. These experiments are designed to provide insights and advancements in the field of object-centric representation learning in general.

4.6 Experimental Setup

In the following chapters (5, 6, 7), we will discuss in detail into various experimental setups centered around the topic of Object-centric Representation Learning, as introduced earlier. This section outlines the rationale and methodology behind our choices in datasets, baseline methods, and evaluation metrics, providing the foundation for the experiments discussed in subsequent chapters.

Datasets The subfield of Object-centric Learning has rapidly evolved, particularly in terms of the complexity of data it can handle. For this research, we selected the synthetic MOVi datasets as our primary data source. There are several reasons for this choice:

First, the MOVi datasets offer multiple variants that progressively scale in visual complexity, making them an ideal testbed for developing and evaluating models across different levels of difficulty. This progression allows us to systematically assess the performance and generalizability of our methods.

Second, the datasets are video-based, which is crucial for our research that leverages temporal continuity—a key aspect in object discovery. The temporal dimension provides rich information that can enhance the learning of object representations over time.

Third, being synthetically generated, MOVi datasets come with rich multimodal annotations that are typically expensive and labor-intensive to collect in real-world datasets. These annotations include detailed object attributes and segmentation masks, enabling comprehensive evaluation across multiple aspects of object-centric learning. This richness is particularly valuable for a nascent field like Object-centric Learning, where diverse data modalities can drive innovation and discovery.

Moreover, the consistency in dataset format simplifies the implementation process, reducing the overhead associated with supporting multiple datasets with varying structures.

Baseline Methods Given the rapid development in this young field, it is challenging to keep up with and reproduce all relevant works, each with different motivations and approaches. To address this, we selected the SAVi family of methods as our primary baseline. This choice was guided by several factors:

First, SAVi methods represent the state-of-the-art in object-centric learning at the time of this research, providing a strong foundation against which to benchmark our methods.

Second, SAVi offers a modular framework where different components—such as slot attention mechanisms, decoder networks, and temporal processing units—play distinct roles. This modularity is particularly advantageous for our experiments, as it allows us to investigate each component separately, facilitating a deeper understanding of their contributions to the overall model performance.

Third, SAVi is specifically designed for video data, aligning perfectly with our focus on leveraging temporal information for object discovery. This video-centric design ensures that our research is grounded in a framework well-suited to the challenges of dynamic scenes.

Task and Evaluation The promise of object-centric learning lies in its potential utility for downstream applications, much like other representation learning methods. However, the most pressing challenge in the field today is the task of unsupervised object discovery. While many object-centric methods have shown success on small-scale, simple scenes, they often struggle with more complex environments, failing to discover and segregate objects accurately.

Motivated by this challenge, our experiments are focused on the task of unsupervised object discovery, with evaluation based on the Adjusted Rand Index (ARI), as discussed earlier. ARI is a robust metric for assessing segmentation quality, making it suitable for evaluating object discovery performance in unsupervised settings.

In practice, the field often reports ARI after removing the background class from consideration, a practice born out of the current limitations where methods tend to mistakenly group background elements, such as shadows, into object segmentation masks. By excluding the background class, the ARI metric—referred to as ARI-FG (Foreground-Only ARI)—provides a clearer, more focused assessment of object segmentation quality, guiding the development of more accurate models. In this thesis, we adhere to this standard practice and report both ARI and ARI-FG metrics to ensure comprehensive and meaningful evaluation of our methods.

Chapter 5

Learning Discrete Object-centric Representations

Building on the challenge of object-centric representation learning introduced in Chapter 4, this Chapter addresses Research Question 3 from Chapter 1 which stated "How can object-centric representation learning approaches, particularly slot-based methods, be designed to capture increasingly complex and abstract visual concepts in a structured manner?". We do so by advancing methods for efficient learning of hierarchical representations via the machinery of discrete representations.

In the context of object-centric representation learning, our goal is to train a neural network capable of discovering and representing complex scenes with multiple independent objects. Each object is associated with a distinct slot, and this collection of slot representations forms a comprehensive representation of the entire scene. Importantly, this representation would allow for efficient manipulation and interaction with objects in downstream tasks. Since the number of objects is different for each scene, current architectures usually define in advanced what is the maximum number of available of slots as a hyper-parameter based on the dataset. Object-centric learning inherently relies on the assumption that visual scenes exhibit a certain degree of independence, manifested via so-called "objects"". These objects can vary in terms of colour, appearance, location, distance, movement, and interactions with other objects. Thus, the desired representation is inherently discrete in nature.

In the broader field of representation learning and deep learning, continuous representations of video over time have traditionally dominated, even in cases where the underlying domain is inherently discrete such as still images or text, for example. However, recent developments in learning discrete representations have shown promise in various aspects, including learning discrete representation for image understanding and generation. These approaches typically quantise representation as the level of individual image patches or audio clips, resulting in a learned codebook

of compressed discrete signals.

We aim to explore the possibility of learning discrete representations of individual objects, rather than focusing solely on discrete representations of image patches, as is the focus of most current works. This is motivated by the similarity to the inherent discreteness of physical objects.

Our experiments in this Chapter will demonstrate for the first time the feasibility and applicability of learning a discrete latent space of object representation.

In the first section of this Chapter, we will provide a brief overview of Vector Quantised-Variational AutoEncoder (VQ-VAE), the foundational method for learning discrete representations, on which our method is built upon.

5.1 Learning Discrete Representations with Vector Quantisation

Originally developed in the context of compression for communications, vector quantisation is a method for mapping between a sequence of continuous vectors into a sequence of discrete digits [93]. The introduction of VQ-VAE by Oord, Vinyals, and Kavukcuoglu [205] sparked a renewed interest in learning neural discrete representations within the context of modern deep learning, employing vector quantisation as a key component.

VQ-VAE is built upon the Variational AutoEncoder (VAE) framework. It can be conceptualised as similar to a VAE but with a quantisation layer as the non-linear operation in its bottleneck layer. In this section, we will offer a comprehensive exploration of VQ-VAE, gradually building upon the fundamental concepts of an AutoEncoder.

This will provide a foundation for a detailed understanding of VQ-VAE and its significance in the realm of discrete representation learning.

5.1.1 AutoEncoder

An autoencoder is a neural network architecture designed for dimensionality reduction and feature learning. It consists of two main parts: an encoder and a decoder.

Encoder: The encoder takes an input data point $\mathbf{x}$ and maps it to a lower-dimensional representation called a "latent space", where a vector in this space is referred to as an "embedding" $\mathbf{z} = \text{Encoder}(\mathbf{x})$. This step is essentially a compression process that captures the most important features of the input data.

Latent space: To encourage the learning of meaningful representations of the data in the latent space, and not just to copy the input, autoencoders are often designed with a bottleneck architecture that reduces dimensionality, which restricts the information that can be communicated from the encoder to the decoder through its latent embedding.

Decoder: The decoder takes the encoded latent embedding and attempts to reconstruct the original input data. The goal is to produce an output that is as close as possible to the input. The loss function is therefore a measure of the reconstruction error between the input and output of the network itself, hence the term "auto" in its name.

Learning: Like most types of neural networks, an autoencoder can be trained end-to-end. The gradient of the reconstruction loss of the output can be fully propagated backward through the latent bottleneck.

5.1.2 Variational AutoEncoder

Traditional autoencoders are effective at dimensionality reduction and feature learning. However, they have a limitation when it comes to generating new data. These models generate deterministic encodings, making them less suitable for tasks like image generation, where we want to sample diverse outputs. Variational AutoEncoder (VAE)s [148] address this limitation by introducing a probabilistic approach to the encoding process.

The key idea behind VAEs is to represent data in a probabilistic manner. Instead of a deterministic encoding, VAEs produce a probability distribution over the latent space for each input data point. This distribution is often assumed to be a multivariate Gaussian.

Encoder: The encoder network of VAE parameterises a posterior distribution $q(\mathbf{z}|\mathbf{x})$ of the latent variable given the input. The posterior distribution of the latent space is usually assumed to be a diagonal Gaussian parameterised by a mean vector μ and a log-variance vector $\log(\sigma^2)$.

Latent Distribution: In addition to the bottleneck design in an autoencoder, VAE's latent representation also incorporate a prior distribution over the latent space. The prior distribution is also assumed to be normally distributed with diagonal covariance, i.e $\mathbf{z} \sim \mathbb{N}(0, 1)$. During training, the distribution of the encoder's output is encouraged to be similar to the prior distribution, typically through a distributional similarity metric like the Kullback-Leibler divergence.

Reparameterisation Trick: Since the VAE's decoder randomly samples a latent from the encoder's posterior distribution during training, the gradient can not flow back through this random sampling process. To address the issue of gradients not flowing through the random sampling process in the VAE's latent space, the reparameterisation trick is applied.

Instead of directly sampling from the posterior distribution provided by the encoder, the trick involves first sampling from a unit Gaussian distribution. The final latent representation is then obtained by scaling and shifting these sampled values using the mean and variance derived from the encoder's output: $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \cdot \epsilon$, with $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. This technique separates the random sampling operation from the gradient flow, allowing end-to-end training of both encoder and decoder.

Decoder: The decoder $p(\mathbf{x}|\mathbf{z})$ takes the sampled encoding $\mathbf{z}$ and generates back to the original input space. Similar to a traditional autoencoder, the reconstruction produced by the decoder is optimised to be as close as possible to the input data through reconstruction error.

Learning VAEs employ a variational inference framework to maximise the likelihood of the observed data given the model. The objective function consists of two key terms: a reconstruction loss that optimises output fidelity and a regularisation term that encourages the latent space to be similar to the prior distribution.

5.1.3 Vector Quantised – Variational AutoEncoder

VQ-VAE is an extension of the VAE approach, designed to capture complex data distributions while providing discrete and structured latent representations. Instead of modelling the latent vector with a continuous distribution like a diagonal Gaussian, the posterior and prior distributions are now categorical. Samples drawn from this distribution will be used as an index for a separate embedding table, or codebook. These indexed latents are then used as input for decoder.

Encoder: The encoder takes input data and produces a continuous latent vector, $\mathbf{z}_e = \text{Encoder}(\mathbf{x})$, similar to a traditional autoencoder.

Vector Quantisation: The continuous embeddings from the encoder are quantised by finding the nearest neighbour in a predefined codebook of discrete latent vectors $\mathbf{e} \in \mathbb{R}^{K \times D}$. The codebook is a set of K vectors, each of dimension D, and is trained in tandem with the encoder and decoder. This process results in a discrete index for each continuous embedding, effectively mapping it to a discrete latent

representation. The posterior distribution is then defined as follow:

$$q(z=k|\mathbf{x}) = \begin{cases} 1 & \text{if } k = \text{argmin}_j \|\text{Encoder}(\mathbf{x}) - \mathbf{e}_j\|_2 \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

Decoder: After quantisation, a code vector is selected from the codebook as $\mathbf{z}_q(x) = \mathbf{e}_k$, where $k = \arg\min_j \|\mathbf{z}_e(x) - \mathbf{e}_j\|_2$. The decoder takes in this latent vector and decodes it back to the original input similar to the traditional autoencoder.

Straight-Through Estimator: Just like the random sampling step in VAEs, the quantisation step in the latent space of VQ-VAE is non-differentiable. To enable the propagation of the learning signal back to the encoder, a technique known as the Straight-Through Gradient Estimator [22] is employed. This method bypasses the quantisation step during the backward pass and directly copies the gradient of the quantised code to the pre-quantised output of the encoder.

Alternatively, other methods such as the Gumbel-Softmax [130] or Concrete distribution [177] can be used to gradually approach the discrete categorical distribution, providing more options for handling the quantisation step during training.

Learning: In VQ-VAE, there are three components to optimised, the encoder, decoder and the codebook. The encoder and decoder can be optimised with the reconstruction loss, using the gradient obtained via the estimator explained above. However, it is important to note that, with the Straight-Through estimator, gradients do not flow through the codebook vectors. To address this, an L2 loss is employed to pull the codebook vector, denoted as $\mathbf{e}_i$, towards the output of the encoder, $\mathbf{z}_e(\mathbf{x})$. Additionally, VQ-VAE introduces a "commitment" term that enforces a strong connection from an encoder output to its corresponding quantised vector in the codebook.

The complete training objective is composed of the reconstruction term (i.e sample log-likelihood), the vector quantisation loss for the codebook and the commitment loss:

$$L = \log p(\mathbf{x}|\mathbf{z}_q(\mathbf{x})) + \|\text{sg}[\mathbf{z}_e(\mathbf{x})] - \mathbf{e}\|_2^2 + \beta\|\mathbf{z}_e(\mathbf{x}) - \text{sg}[\mathbf{e}]\|_2^2 \tag{5.2}$$

where "sg" stands for the "stop gradient" operation that prevent gradient flow through the term that the sg operation is applied on.

In Table 5.1 we provide a brief summary to highlight how the modelling differs between AutoEncoder (AE), VAE and VQ-VAE.

Table 5.1: Comparison of components between Autoencoder (AE), Variational Autoencoder (VAE) and Vector Quantised - Variational Autoencoder (VQ-VAE).

Component	**AE**	**VAE**	**VQ-VAE**
Encoder	Maps input to continuous embeddings	Maps input to mean and variance for Gaussian distribution in latent space	Maps input to continuous embeddings
Latent Space	Continuous, unstructured	Continuous, often Gaussian distributed	Discrete, structured by codebook
Decoder	Maps encoding to output	Sample latent and decode	Map discrete indices to codebook vector and decode
Losses	Reconstruction Loss	Reconstruction Loss + Latent regularization	Reconstruction Loss + Vector Quantization Loss + Commitment Loss
Optimisation	Backpropagation end to end	Reparameterisation trick	Straight-Through Gradient Estimator

5.2 Related Work on Learning Discrete Representations

Having presented a high level overview to give an understanding of the mechanics of VQ-VAE, this section now delves into an examination of related works that leverage this concept. Broadly, the body of literature that explicitly deals with learning or utilising a discrete representation can be categorised into two main groups.

The first group described in subsection 5.2.1 employs quantisation as a mechanism to compress input signals, which previously took the form of image patches or audio clips. This approach focuses on the efficient compression of data. The second group described in subsection 5.2.2 is motivated by the use of discrete representations as a means of communication between different neural modules, with a focus on enabling interaction and information exchange within a neural network architecture. Our work in this chapter bridges between the groups, aiming to learn discrete representations of "objects", instead of image patches, that are intended to be more suitable for processing by downstream modules.

5.2.1 Discrete Signal Compression

The concept of VQ-VAE, originally introduced by Oord, Vinyals, and Kavukcuoglu [205] provides a comprehensive framework for learning discrete latent representations in the context of modern deep learning. Initially applied to the task of generating visual representations for images, this approach has found widespread adoption as a foundational component for downstream tasks and has extended its applicability to other domains, including audio representation.

To mitigate the challenges of learning representations for complex visual signals, VQ-VAE adopts a two-step process. It starts by dividing an input image into non-overlapping patches. Each of these patches is then independently encoded, quantised, and decoded using the vector quantisation framework outlined in Section 5.1. Consequently, each image becomes represented by a grid of indices, where each index corresponds to a code vector within the learned codebook. This approach, due to its patch-based nature, tends to yield individual codes that capture simple visual concepts.

Vector-quantised models are known for their instability during training, where issues such as codebook collapse or under-utilisation, and sensitivity to hyperparameters such as codebook size, can be present. Roy et al. [230] has undertaken research to address these challenges, exploring various aspects to improve VQ-VAE training. This includes the utilisation of soft expectation maximisation (EM) and the fine-tuning of the codebook size to better align with target tasks.

Takida et al. [265] mitigated codebook collapse by incorporating a technique that involves adding Gaussian noise during training, which is annealed over time, to the encoder output. This approach has proven effective in enhancing the stability of VQ-VAE training.

Additional efforts to enhance the training dynamics of VQ-VAE have been introduced by Huh et al. [125]. Their contributions include a novel codebook reparameterisation, the application of alternating optimisation strategies, and improvements to the commitment loss function.

To enhance the visual fidelity of generated images, Esser, Rombach, and Ommer [76] introduced VQ-GAN, which incorporates a Generative Adversarial Network (GAN) loss applied to the generated visual patches. This addition improves the quality of the generated images by introducing adversarial training. Furthermore, Yu et al. [304] took this approach a step further by scaling up VQ-GAN and integrating it with a Vision Transformer backbone. In addition to scaling, they introduced various techniques to improve codebook utilisation and learning. These techniques include projecting codes to a lower-dimensional space and normalising them before the quantisation lookup step.

In the audio domain, Baevski, Schneider, and Auli [12] applied the vector quantisation technique to learn audio speech representations. This approach demonstrates the versatility of vector quantisation in capturing meaningful representations in the audio domain. Similarly, Dhariwal et al. [63] employed vector quantisation for the generation of music. To enhance the training dynamics, they introduced a multi-scale variant of VQ and a "random restarts" technique to mitigate codebook collapse during training. This involves replacing low-usage codebook entries with the encoder outputs, ensuring a more stable and effective training process in music generation.

Most Relevant to our work in this thesis is SLATE [247], an approach that also aims to harness discrete visual representations of image patches for learning object-centric representations. This approach utilises a pre-trained discrete encoder and decoder, employing them as the target for prediction tasks.

However, it is important to note that, like all of the methods discussed in this sub-section, the focus remains on utilising vector quantisation to learn discrete representations from patches of the input signal, whether those are image patches or audio clips. In contrast, the approach presented in this chapter explores the possibility of learning discrete representations at a higher level of abstraction, specifically at the object-level.

5.2.2 Discrete Latent Communication

A line of research that shares a similar motivation with our approach, which focuses on learning discrete object-centric representations, is the exploration of using discrete latents as a means of communication between different modules within a neural network.

As introduced in Chapter 4, several researchers advocate for neural networks to comprise multiple modules, each with distinct architectures and characteristics. This perspective is particularly relevant for models with algorithmic execution [273] or reasoning capabilities [17, 18, 89, 90]. In a manner similar to the encoder and decoder components in an autoencoder, future neural networks may encompass modules such as perception, abstraction, planning, and goal setting [159].

Efficient communication among these components ideally requires them to share a common language, manifested as a shared representation space. Explicitly constraining the communication language to utilise a common codebook can serve as a valuable inductive bias.

Liu et al. [168] posited that the use of discrete symbols serves to limit the bandwidth of communication. This limitation results in reduced complexity for representations that need to be learned and synchronised across modules, making the learning process more manageable. Furthermore, the use of an explicit codebook en-

ables the reuse of previously learned symbols. Reusing these components in various combinations promotes systematic generalisation in new situations and facilitates the exchange or update of one component for another when confronted with new out-of-distribution (OOD) settings.

In their experiments, Liu et al. [168] applied quantisation to various components of a Graph Neural Network [151], a Transformer architecture [272], and a Recurrent Neural Network [91]. They observed improvements in generalisation when using discrete representations. Furthermore, Liu et al. [167] extended this work by introducing an adaptive quantisation bottleneck conditioned on the input. This extension achieved better performance in tasks related to visual reasoning and reinforcement learning.

From a neuroscience perspective, various areas in the brain, including the hippocampus [287], have shown an adaptation to discrete variables, such as concepts, actions, or objects. This observation suggests that there might be an evolutionary advantage to utilising discrete encoding. Such an encoding approach may partially explain the remarkable generalisation capacity observed in the brain, which often surpasses that of current neural networks.

As the goal of learning object-centric representations is to provide a set of compact, independent representations for further downstream modules, our work of learning discrete object-centric representations can be considered as a direct extension of this line of work.

5.3 Methods

Following the review of the contributions of other researchers to the development and state of the art in AutoEncoders, Variational AutoEncoders and Vector Quantised Variational AutoEncoders, in this section, we introduce Vector Quantised - Slot Attention on Video (VQ-SAVi), our own approach for acquiring discrete object-centric video representations. Our method is grounded in the VQ-VAE (Vector Quantised Variational Autoencoder) framework, as detailed earlier in this Chapter, in Section 5.1.

Distinguishing itself from approaches focused on attaining discrete representations at the image patch-level, VQ-SAVi builds upon the recent (2021) object-centric methodology proposed by Conditional Object-centric Learning on Video (SAVI) (Slot Attention for Video) [150]. This choice is made to showcase the efficacy of learning discrete representations within an object-centric paradigm. A more comprehensive understanding of SAVi and its positioning within the broader object-centric methods landscape was presented in the previous chapter, in Section 4.4. We now describe the 6 steps for this method, covering the visual encoding, slot initialisation,

corrector, predictor, quantiser and finally the decoding stages.

In Figure 5.1, we illustrate the overall architecture for our object-centric learning pipeline, based on the SAVi [150] baseline (the top figure) and our two Vector Quantised variants in the middle and bottom. For the first frame, we extract the ground truth bounding boxes information of each object in the scene to initialise the object slots representations. The visual encoder then process the raw RGB input frame into a spatial feature map. The corrector module implements the Slot Attention mechanism, with the intialised slots as the queries, and the visual features map is independently projected into the keys and values. A slot-wise gated recurrent unit further updates the slot representations. The predictor then further transformed the object representations by facilitate the exchange of information between slots. Finally, the slot decoder predict the target optical flow signal from the individual slots. Along the way, we can quantise the slots after the Corrector (middle) or after the Predictor (bottom). More details of each individual components is presented below:

Visual Encoder: To initiate the processing of visual input and obtain a feature map for it, each video frame x_t at timestep $t \in 1, \ldots, T$ undergoes encoding through a compact CNN network. This network incorporates non-linear ReLU activation functions between layers, generating a feature map $h_t = \text{Encoder}(\mathbf{x}_t) \in \mathbb{R}^{h \times w \times d}$ as the output.

In order to preserve 2D positional information for subsequent modules, we introduce linear positional embeddings to each vector in the feature map. Subsequently, each feature vector undergoes transformation through a compact Multi-Layer Perceptron (MLP). The resulting set of visual features is expressed as:

$$\mathbf{h}_t = \text{MLP}_\text{e}(\text{Encoder}(\mathbf{x}_t) + \text{PosEmb}(h, w))$$

For all experiments detailed below, we follow prior works and employ a CNN with 5 convolutional layers. The convolutional layers utilise a kernel size of 5×5, a stride of 1, and feature dimensions set to $d = 32$. The final MLP comprises a single hidden layer with a size of 64 and an output dimension of $D = 128$.

Slot Initialisation: To completely decouple the visual features from the slot representations of objects, a distinct set of parameters is initialised for the slot representations. Let $S_t = [\mathbf{s}_t^1, \ldots, \mathbf{s}_t^K] \in \mathbb{R}^{K \times D}$ denote the set of K slots, where each slot ($\mathbf{s}_t^k \in \mathbb{R}^D$) can represent a distinct object in the scene.

For all our experiments, we employ conditional initialisation based on the bounding boxes from the first frame. For every object appearing in the initial frame of

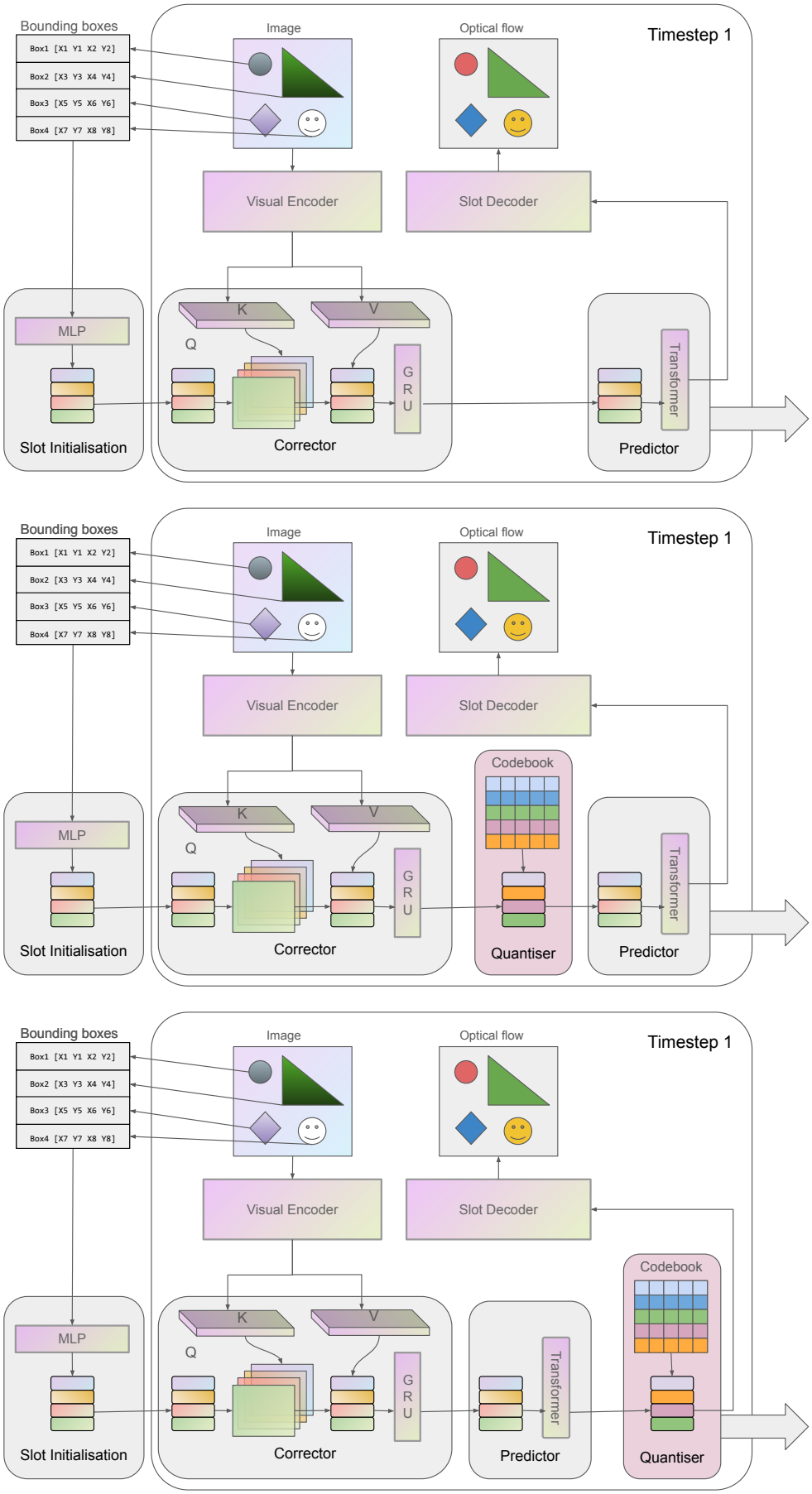

Figure 5.1: Architecture of the baseline SAVi (top) and our two variants VQ-SAVi with quantisation applied on the corrector (middle) or on the predictor (bottom) output.

the video, we train a simple MLP to project the bounding box coordinates of each object to a vector of dimensionality $D = 128$. To condition the first slot to represent the background, a dummy value of 0 is always prepended as the first bounding box. In the event of fewer objects than the designated number of slots K, a dummy value of -1 is added to the remaining slots. The initial conditional slot representations are then formulated as $S_0 = \text{MLP}_i(\text{bboxes}_0) \in \mathbb{R}^{K \times D}$.

Corrector: Given the initial slot representations S_{t-1}, the corrector module aims to "correct" the previous slots with information provided by the encoded visual features $\mathbf{h}_t$ in the current timestep. We implement the corrector using a Slot Attention module taken from [170], where the previous slots serve as queries and the visual features as keys and values. The attended outputs are further updated via a Gated Recurrent Unit: $\hat{s}_t^k = \text{GRU}(\text{SlotAttention}(S_{t-1}, \mathbf{h}_t))$.

To enhance the expressiveness of the module, each slot is then normalised using Layer Normalisation [10] and transformed with an MLP in a residual branch: $\hat{\mathbf{s}}_t^k = \hat{\mathbf{s}}_t^k + \text{MLP}_c(\text{LN}(\hat{\mathbf{s}}_t^k))$.

Predictor: While the corrector organises and updates visual information for each object slot, the predictor facilitates the interaction and exchange of information between object representations.

Following SAVi, we implement the predictor as a simple, 1-layer Transformer [272] with the original scaled dot-product multi-head self-attention layer (MHSA). Similar to previous modules, we further transform the output with an MLP, incorporating normalisation and a skip connection before each operation.

The predictor predicts the slot representations for the next timestep as:

$$S_{t+1} = \text{LN}(\text{MLP}_p(\tilde{S}_t) + \tilde{S}_t)$$

where

$$\tilde{S}_t = \text{LN}(\text{MHSA}(\hat{S}_t) + \hat{S}_t)$$

In our experiments, we use 4 heads for the self-attention layer, with a dimensionality of 128 for the queries, keys and values projection and an MLP with a 256-dimension hidden layer.

Quantiser: The quantiser serves as a crucial component in the process, converting continuous object features into an index corresponding to a set of discrete codebook features. This step is integral to our overarching objective of learning a discrete object-centric representation. In alignment with the Vector Quantisation - Variational AutoEncoder (VQ-VAE) framework, the quantiser operates by associating

each continuous feature vector with the nearest prototype vector in a predefined codebook. This process inherently induces a form of data compression, as each feature is now represented by its corresponding discrete index in the codebook. Let $C = [\mathbf{e}_1, \ldots, \mathbf{e}_K]$ denote the codebook, where $\mathbf{e}_k \in \mathbb{R}^D$ represents the k-th embedding vector. The quantiser function q for a given continuous slot representation $\mathbf{s}$ is defined as:

$$q(\mathbf{s}) = \arg\min_k \|\mathbf{s} - \mathbf{e}_k\|^2$$

In practical terms, this results in replacing each continuous feature $\mathbf{s}$ with its corresponding discrete index z, such that $\mathbf{s} \approx \mathbf{e}_z$. This discrete index z becomes a fundamental element in the subsequent stages of our model, aiding in the learning of object-centric representations.

The codebook C embeddings and the associated mapping to indices z are jointly optimised during the training process, allowing the quantiser to maintain and adapt to the characteristics of the input data.

To improve the expressiveness of the discrete representation, we borrowed the multi-head approach from the transformer for the quantisation step. Each object representation $\mathbf{s}_t^k \in \mathbb{R}^{128}$ is split into 8 heads $\mathbf{s}_t^k \in \mathbb{R}^{8\times16}$. We build a codebook of 256 embeddings of 16 dimensions each $C \in \mathbb{R}^{256\times16}$ and perform the quantisation in parallel for each head. We also set the hyperparameter for the commitment loss weight as 0.25 of the reconstruction loss.

Decoder: To encourage the learning of object-centric representations, in the reconstruction step we employ a slot-wise Spatial Broadcast Decoder [283]. The decoder functions independently for each slot, decoding both the 2D target signal and an alpha mask, the latter serving to quantify the contribution of a slot representations at each decoded location.

For each slot k, the decoder output is denoted as y_k^t, representing the decoded target signal, and $\hat{m}_k^t$, the alpha mask. The final reconstruction at timestep t, denoted as $\mathbf{y}_t$, is a combination of the decoded targets of each slot, weighted by the normalized alpha masks of the respective slots:

$$\mathbf{y}_t = \sum_{k=1}^{K} m_k^t y_k^t, \quad m_t = \text{softmax}_K(\hat{m}_k^t), \quad \hat{m}_k^t, y_k^t = \text{Decoder}(s_{\hat{k}}^t).$$

Here, m_k^t represents the normalised alpha mask for slot k at timestep t, obtained through the softmax function. The alpha mask $\hat{m}_k^t$ and the decoded target y_k^t are generated by the decoding function Decoder applied to the slot representations $s_{\hat{k}}^t$, where $\hat{k}$ is the discrete index obtained from the quantiser.

In our experiments, to reduce the computation and memory requirements, each slot is first broadcast to a smaller grid of size 8×8 before being up-scaled to the target size through a series of 5 transposed convolutional layers with a kernel size of 5×5 and stride 2.

This slot-wise decoding mechanism allows for the reconstruction of the final target while emphasising the distinct contributions of individual object-centric representations.

Discrete slot representations: To obtain discrete object representations, we introduce two distinct variants—**Variant A** involves quantising the slots immediately after acquisition by the Corrector, while **Variant B** quantises them post the Predictor's output. The rationale behind these variants stems from the hypothesis that discretising object-centric representations at different stages may yield different effects on our learned representations.

In **Variant A**, the Corrector module parses visual information from the image encoder and updates the object slots from the preceding timestep. This approach potentially introduces a bias towards grouping based on visual information, as the slots are influenced by the immediate visual context.

Conversely, **Variant B** employs quantisation after the Predictor module's output. The Predictor module facilitates the exchange of information between slots, allowing for the learning of their interactions. Quantising the Predictor output may encourage the model to focus more on these interactions, potentially leading to a refined understanding of object relationships and dependencies.

The architectural depiction of the continuous baseline, along with the two discrete variants, is illustrated in Figure 5.1.

5.4 Results

In this section, we present a comparative analysis of the results obtained from learning discrete object-centric representations. We evaluate the performance on two variant datasets, namely B and C, of the MOVi multi-object datasets which are outlined below. For both datasets, we opt to use optical flow as the decoding target instead of RGB pixel values, a choice motivated by the challenging nature of the datasets and the potential for optical flow to capture dynamic object interactions more effectively. For a more detailed overview of the datasets, please refer to Chapter 4.

Optimising Codebook: Vector Quantised models are notoriously unstable, sensitive to hyperparameter selection and random initialisation. We observe the same

challenges in applied vector quantisation for object-centric methods. A common root cause for this issue is the collapse or low-usage of the codebook entries, in which the discretisation process maps all input embeddings to only a small subset of codebook indices. The Perplexity metric of a discrete distribution ($PP(q)$) is commonly used in the vector quantisation literature [205] as a proxy for the number of codebook entries used and is computed using the following formula:

$$PP(q) = \exp(H(q)) = \exp\left(-\sum_{c=0}^{|C|} \mathbf{q}_c \cdot \log(\mathbf{q}_c)\right)$$

Here, $\mathbf{q}$ is vector which contains of the proportion of usage of the entries in the codebook. Perplexity comes from information theory, which is essentially the exponential of the entropy $H(\mathbf{q})$ of the distribution q. By its definition, perplexity ranges from 1 to the number of entries in the codebook $|C|$. The resulting perplexity value provides a measure of how well the codebook represents the input data with higher perplexity values indicating a more diverse and effective use of codebook entries.

End to end learning of the model and the codebook using the original Vector Quantisation formulation is challenging. This is due to interaction between simultaneously updating the encoder outputs to match the codebook entries, while also updating the codebook entries themselves. All of our initial experiments in this chapter resulted in a collapse of the discrete representation in the codebook, as indicated by perplexity value of 1 i.e. the network failed to perform the autoencoding task.

To encourage better utilisation of the codebook and prevent collapse at the early stages of learning, we then tried initialising the codebook values using the K-means clustering method. After the network is randomly initialised, we passed the entire training set through the untrained network and recorded all the encoder latent vectors i.e. the input for the quantiser. We then perform K-means clustering on the latent vectors of all inputs, with $K = |C|$ as the number of codebook entries and then initialise the codebook using the K clustered centroids. Despite this effort, end to end learning using the K-means initialisation approach did not help and also resulted in codebook collapse.

Through multiple refinements guided by the above initial experimentation, and the literature [168, 180, 205], the following steps emerged as those providing the best possibilities in terms of performance.

We obtain a working recipe using the following combination of techniques:

- $L2$-normalisation of the codebook and input encodings [180]. This in effect converts the Euclidean distance when quantising to cosine distance.

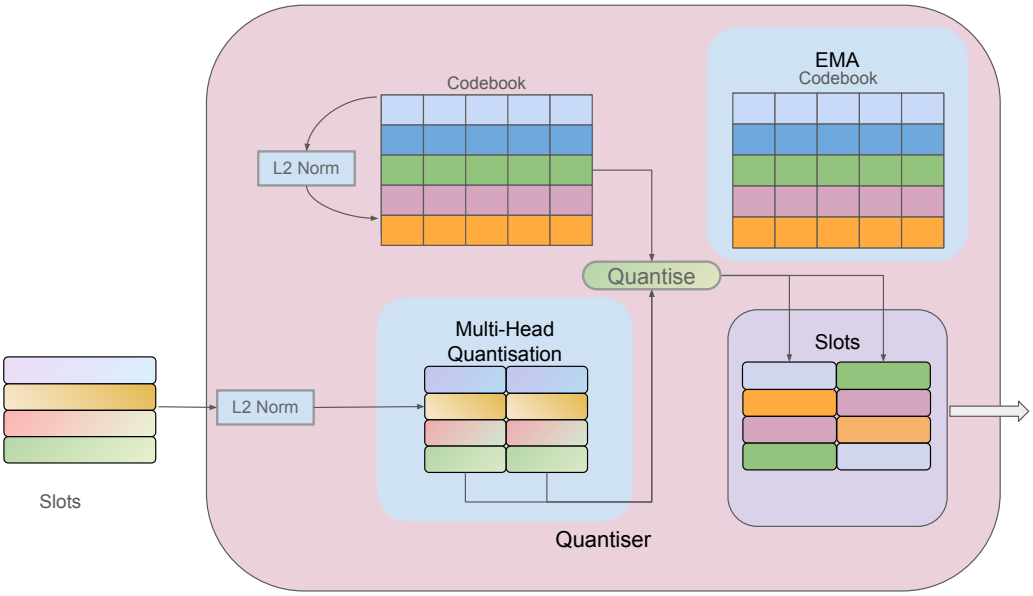

Figure 5.2: Detailed diagram of our quantiser module, consists of: a) L2-normalised codebook's vectors and slots encoding prior to quantise, b) Multi-head quantisation along the channel dimension and c) Update the codebook via Exponential Moving Average mechanism of past encodings.

- Multi-head Vector Quantisation [167]. Inspired by the multi-head architecture in the attention layer of the Transformer model, we split our latent vector into smaller chunks and perform quantisation on those chunks in parallel "heads". We use 8 heads for all of the following experiments.

- Exponential Moving Average Codebook [205]. This alternative formulation of VQ removes the quantisation loss and updates the codebook entries via an exponential moving average of the past encoding quantised to each codebook vectors. We found the exponential weight $\gamma = 0.9$ works well for all experiments.

The MOVi-B dataset: Figure 5.3 illustrates the progression of the unsupervised object-discovery metric ARI-FG in both the training and validation sets during training steps. On the training set, we observe that both Vector Quantised variants reach a comparable level to the continuous SAVi baseline. This shows that despite a strong inductive bias of discreteness imposed by Vector Quantisation, we are still able to learn to discover objects similar to the state of the art. However, a small performance gap exists on the validation set where the continuous SAVi version performs consistently better as indicated by the learning progression. It suggests that while the discrete representation captures discovered objects effectively, there may be challenges in fully representing the complexities of object-centric structures.

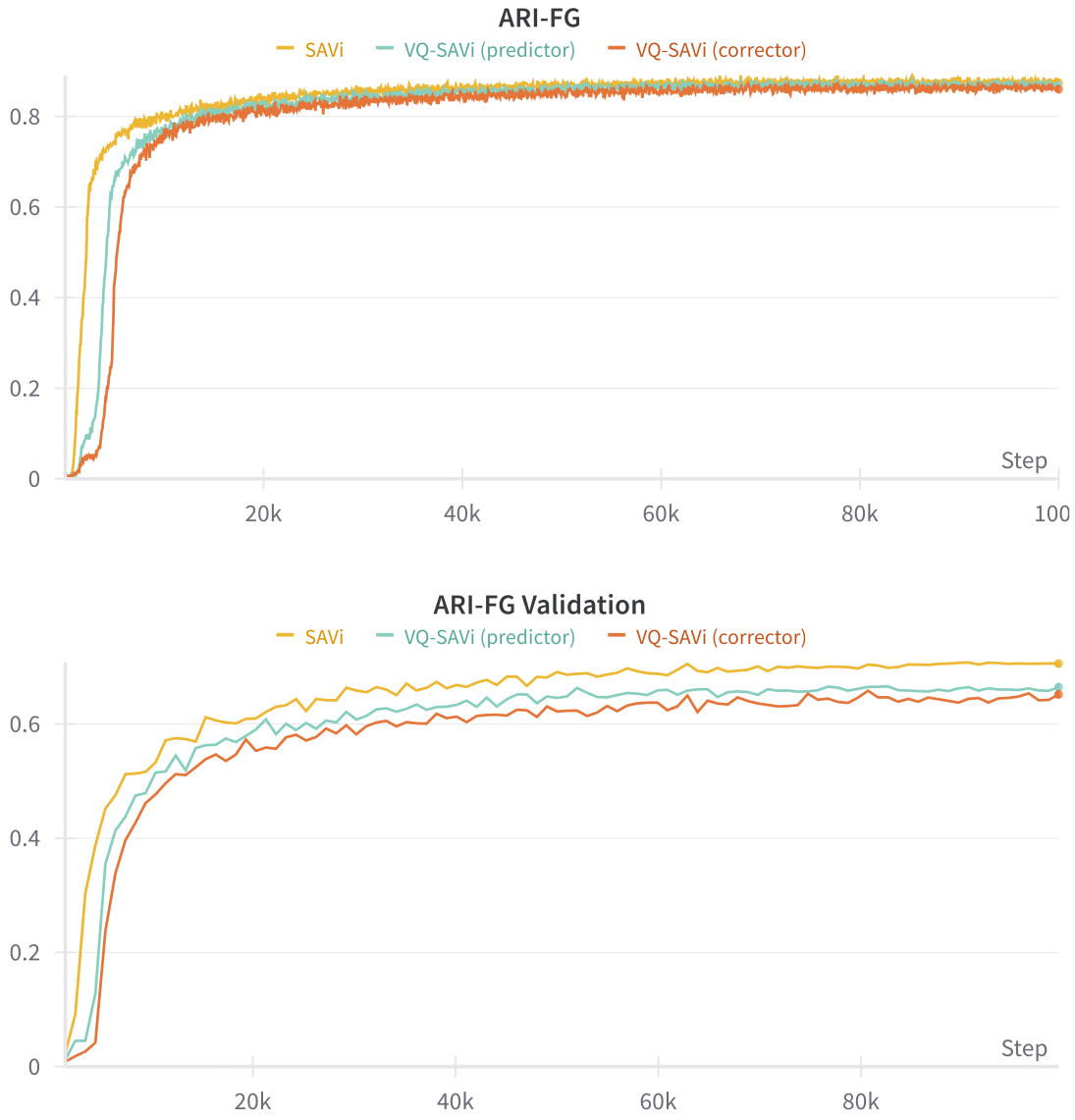

Figure 5.3: The ARI-FG metric (along the vertical axis) evaluated on the training (top) and the validation sets (bottom) over the training steps (along horizontal axis) of the baseline SAVi method and our Vector Quantised variants (VQ-SAVi Predictor and Corrector) on the MOVi-B dataset.

Comparing the quantisation of object slots after the corrector vs. after the predictor module, there is indication that quantising after the predictor enables the model to learn faster and achieve slightly higher performance than quantising after the corrector. This might partially be explained by the effective use of the codebook as indicated by the perplexity metric values.

In Table 5.2, the perplexity values for the experiments on the MOVi-B dataset confirm that quantising after the predictor achieves a higher average codebook usage compared to quantising the slot representations after the corrector. This means that

Table 5.2: Comparisons between SAVi and our VQ-SAVi variants on the MOVi-B dataset.

Method	ARI-FG training	ARI-FG validation	Perplexity
SAVi	0.874	0.706	-
VQ-SAVi corrector	0.860	0.652	57
VQ-SAVi predictor	0.868	0.664	71

quantising the corrector on average uses 57 codebook entries out of 256 while quantising the the predictor output uses 71 entries over the entire dataset. This suggests that the quantisation process after the predictor module results in a more diverse and effective use of codebook entries. These results highlight the competitiveness of our Vector Quantised variant with the baseline SAVi method and the importance of the choice of the quantisation placement process in the model architecture.

The MOVi-C dataset: MOVi-C is a substantially more challenging dataset in terms of visual complexity while retaining similar motion dynamics compared to MOVi-B. Instead of geometric shapes with uniform colour for both the objects and background, MOVi-C is rendered from 3D-scans of real-world objects with complex textures. All objects randomly move into the scene with the background also using real images. On this dataset, we encounter limitations in our current vector quantised approaches, as can be seen by the relatively lower performance compared to SAVi.

Figure 5.4 shows the progression of the ARI-FG metric over the course of training for our baseline and the two quantised variants. The gap in the ARI-FG metric values on the training and validation sets starts to widen when compared to the continuous baseline, with the difference value of 0.05 and 0.1 respectively.

Table 5.3: Comparisons between SAVi and our VQ-SAVi variants on the MOVi-C dataset.

Method	ARI-FG train	ARI-FG val	Perplexity
SAVi	0.8190	0.6179	-
VQ-SAVi corrector	0.7662	0.5202	90
VQ-SAVi predictor	0.7687	0.4962	75

Examining the values for the perplexity metric for MOVi-C in Table 5.3, we observe an increase in the average usage of the codebook, as expected when using a more challenging dataset. Contrary to the results on MOVi-B subset, here we

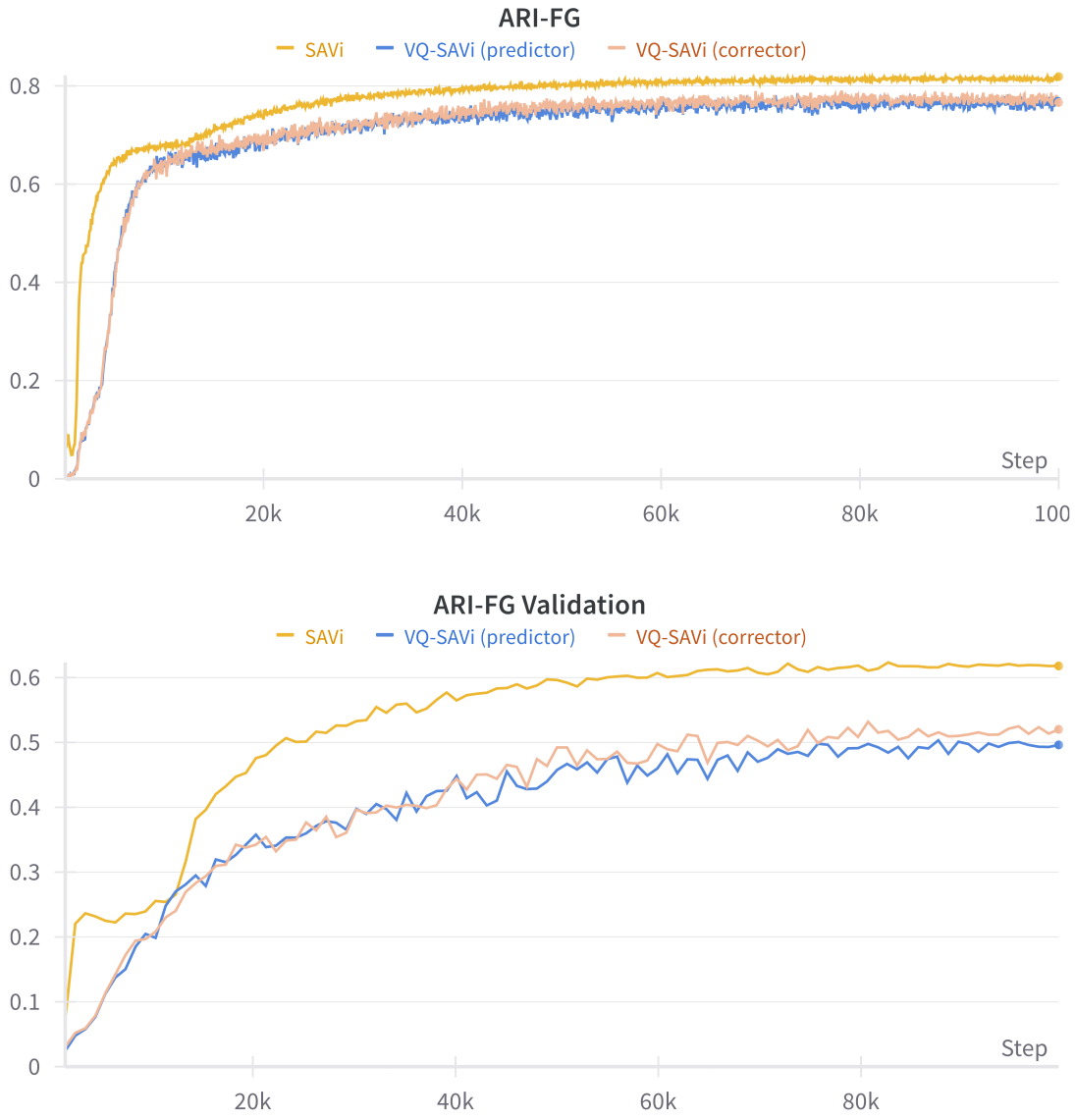

Figure 5.4: The ARI-FG metric (vertical axis) evaluated on the training (top) and the validation sets (bottom) across train steps for the baseline SAVi method and our Vector Quantised variants on the MOVi-C dataset.

saw the variant whereby we quantise the slot representations after the corrector yields a higher perplexity value than the predictor variant. One possible explanation for the difference could be in the nature of the information retained in the slots after the corrector and predictor. The corrector's main goal is to capture visual information about an object while the predictor will model their interactions by predicting their state in the next timestep. While the MOVi-C dataset is composed of more challenging visual scenes and objects, resulting in higher codebook usage of the corrector from 57 to 90, the objects' movements and interactions from MOVi-B to MOVi-C are roughly similar, as indicated by the similar perplexity of the predictor

variant of 71 and 75 respectively. However, for both datasets, given the codebook size of 256, the average usage under 100 indicates a relatively low codebook usage.

5.5 Discussion

In this chapter, we investigated a novel direction of learning discrete representations within the realm of object-centric video representation. Extending previous works that employed Vector Quantisation for representation learning, our focus is on acquiring a higher-level and more abstract representations at the object level. We showcased the feasibility of our approach by integrating it into a state-of-the-art object-centric representation learning method designed for video datasets. We focus on the unsupervised object-discovery task and measure the performance with the unsupervised segmentation metric ARI-FG following the standard literature.

In this initial exploration, we established the competitiveness of our novel approach on a simpler dataset MOVi-B, while also acknowledging the current limitations when scaling to a more challenging dataset, MOVi-C. While the baseline and our quantised variants are overfitted as exhibited via the gap in between training and validation metric values, our discrete object-centric representation increases this generalisation gap. As seen with the challenge in optimising the quantiser, collapse or under-utilisation of codebook could potentially be the root cause of this problem. This prevents the model to take advantage of its full capability and hampers its ability to generalise to unseen samples. A better quantisation technique for representation learning [185] could potentially help alleviating the low codebook usage problem.

While we demonstrate that it is possible to learn discrete object representation on par with continuous counterparts in terms of object discovery and learning efficiency, a potential benefit of discrete representations could be its evaluation in downstream tasks or in out-of-distribution generalisation capabilities. Exploring the advantages of this discrete object representation in out-of-distribution generalisation constitutes an interesting avenue for future work.

Learning discrete object-centric representation also grants us access to a novel object codebook. Exploring the structure or incorporating this codebook elsewhere could lead to novel applications in object interaction and manipulation in the domain of robotics.

Another possible direction for future work could involve the combination of visual quantiser and object quantiser, potentially offering enhanced capabilities and richer representations.

In this chapter, we attempted to address our Research Question 3 by focusing on a novel discrete representation format for object-centric learning. While we demon-

strate the viability of learning discrete object-centric representations in this chapter, our novel representation format, unfortunately, does not appear to significantly contribute to advancing the overarching goal of scaling the object-centric representation learning method to large-scale real-world datasets. Beyond the common challenge of optimising discrete representations, it becomes evident that the performance bottleneck hindering the scalability of the current state-of-the-art pipeline for unsupervised object learning and discovery likely resides in other components. Currently, our object-centric method still relies on visual features from a relatively small CNN backbone, where target reconstruction relies on extra optical flow signal, and object decoding only interact at the pixel level. Much like how discrete representation helps to advance visual representation learning at scale, our approach to learning discrete object-centric representations could be revisited when object-centric learning methods have advanced to a better scale.

In the following Chapter 6 and in Chapter 7, we shift our focus to other components, namely the object-centric decoder and the roles played by the data and features used for object-centric learning. By delving into these aspects, we aim to pinpoint and address the specific challenges that hinder scalability, offering insights and potential solutions that can pave the way for the effective application of object-centric representation learning methods on larger, more complex, real-world datasets.

Chapter 6

Attentional Slot Decoder

As discussed in Chapter 4, an important goal of object-centric learning is to automatically obtain a set of object representations for visual content that are independent, compatible and complete. As a universal function approximation, a neural network as simple as a MLP can learn to approximate any function, given enough hidden units and input-output pairs of the function being approximated [122]. These input-output pairs directly supervise the training process towards approximating the correct function. For unsupervised pre-training where the training data pairs of the pretext tasks are not what we ultimately want, this implies that inductive biases are crucial for steering the model to learn useful representations. Unsupervised discovery of constituent objects in a scene, and learning their representations, therefore must requires some form of inductive biases. This inductive bias can be manifested in any of its architectural components such as the visual encoder, the object representations bottleneck or the latent decoder.

In this chapter we focus on the design of the decoder module in object-centric representation learning methods, with the goal of addressing Research Question 4 in improving the efficiency of Slot-based Object-centric learning method. Besides a number of methods that use contrastive learning [149, 174], the majority of object-centric learning models use the framework of generative and autoencoding as the pre-training tasks, which requires representations to be decoded back into the input space. The design of the object decoder thus has an integral and major role in many past and future object-centric learning methods.

In the current literature, all decoders for object-centric learning methods can be categorised into two groups, slot-based independent decoding or set-based joint decoding. Each decoding approach has its own set of advantages and disadvantages, where some promote independence and compatibility of slots but are less powerful and require more compute time and memory during training, while others are more powerful but lack the inductive bias that is helpful for learning object-centric representations.

The main contribution of this chapter is the proposal of a very simple and straightforward decoder for object-centric learning. Our approach, termed "Slot Attention decoder" uses an attention mechanism to allow rich interaction between slots in the latent space, while the more expensive visual decoder only needs one iteration to reconstruct back the high-dimensional input. The experiments show that this simple approach is faster and requires less memory and compute time to learn than other slot-based decoding approaches in term of reconstruction error, while still achieving similar performance on the unsupervised object discovery metric ARI-FG.

6.1 Related work

In this section, we briefly review the existing approaches on designing a decoder for object-centric methods. Before that, we discuss the goal and desired characteristics of such a decoder.

6.1.1 Desiderata: Ideal Characteristics of a Slot Decoder

In standard architectures for computer vision problems, the representation of an image or video is usually a latent vector that summarises the information from the entire scene or clip. These latent scene representations are often pooled from a spatial grid of intermediate representations that are downscaled from the input's initial spatial dimensions. In other architectures like the Vision Transformer [67] family, the scene representation can also be designated beforehand as a special token and is learned jointly through the attention mechanism with other spatial tokens. In addition to the scene representation, the decoder can also leverage other intermediate representations with spatial dimensions to help reconstruct high-frequency information in the input, i.e the U-Net architecture [228].

Object-centric representation learning, on the other hand, will typically yield a **set** of latent representations without any explicit spatial dimensions, each corresponding to an object in the scene. We would like this set to be a complete, independent and interchangeable set of object-representations. That is to say the set of all object-representations together are able to describe the whole input scene, while each latent vector describes an independent object using a similar format that can be used interchangeably in downstream models. These independence and interchangeability properties are thus usually integrated into the pre-training autoencoding task throughout all the different modules of the entire architecture, from encoding to decoding.

Here, we discuss the most pertinent properties of an object representation decoder, and their implications on the latent representation and on the entire system.

Completeness: A good object representation decoder must be first and foremost a good scene decoder for the autoencoding pre-training task. From the set of latent vectors, the decoder should be able to reconstruct the input scene including all the objects and their backgrounds.

Powerful: Even though reconstructing an input is not what we ultimately desire from an object-centric representation, one cannot learn a good set of representations while performing poorly on the autoencoding pretext task which is explained earlier in Chapter 4. A powerful decoder could potentially help to scale object-centric learning methods to more diverse, complex and real datasets.

Efficient: Except for the recent class of iterative denoising diffusion models [119], most generative approaches are designed to generate an output scene with a single pass from a latent vector. Naively applying these methods to individual object representations would require a decoder step for each vector in our set. This can make training and evaluating the model very expensive in term of compute time and memory requirements, especially when the number of latent vectors is large, i.e it is built from a complex dataset with many small objects.

Permutation invariance: During training, an object in a scene can potentially be factored into any of slots in the set. Since we want to maintain a degree of compatibility between object latents to be able to use them for some downstream tasks, a decoder should ideally be permutation invariant. That is for any permutation of the latent vectors in the set, the decoder should still be able to reconstruct the same scene.

The above four criteria of an object-centric decoder can sometimes be in conflict with each other. For example, an independent slot-based decoder like the Spatial Broadcast Decoder [283] inherently promotes compatibility among slots but this can be computationally expensive to achieve while being more restrictive and not as powerful as other methods. On the other hand, a set-based decoder like the Transformer Decoder [272] can be more efficient to train, allowing for richer interactions between latent vectors, but does not enforce object-independence among slots.

We now review the two main approaches in designing a decoder for object-centric learning methods, based on the criteria discussed above.

6.1.2 Slot-wise Decoder

The first approach to designing a decoder, "slot-wise decoder", is sometimes referred to in the literature as a "mixture of component decoder"[28, 138]. The distinctive characteristic for this name stems from the fact that the final reconstruction output

of the decoder is a weighted mixture of multiple reconstructions, where each reconstruction is decoded from using a single slot in the set of object representations. Following that, the final scene can be composed from all the mixture components depending on the slots' weight. Consequently, for one training sample, the decoder needs to process more forward and backward passes proportional to the number of slots. While the decoding of slots of a sample can be done in parallel, the decoder still needs to process more forward and backward passes in total. This increases the amount of compute time and memory required with a multiple of the number of slots, which is a hyperparameter set before training usually in the range from 5 to 30 depending on the dataset. One can tradeoff between the number of latent vectors and the dimensions of each latent vector in order to maintain a similar capacity in the latent space. For example, instead of taking a single CLS-token from a Vision Transformer of dimension 768, one can have a set of 8 object representations, each having a dimension of 96.

The original motivation for this approach, and also part of its strong advantage, is the baked-in independence assumption between object-representations learned directly during training. Since the same decoder module is used for all slot vectors, they are all encouraged to be compatible directly from the pre-training stage.

Most methods in this category use a form of Spatial Broadcast Decoder [283], where a slot representations is broadcast to all spatial locations, then concatenated with positional information.

6.1.3 Set-based Decoder

While the ultimate goal of object-centric learning is not to be able to faithfully reconstruct the input, the ability to do the autoencoding pretext task still affects the representation learning challenge. Better autoencoding skill would also allow the methods to scale to more complex and challenging real-world datasets.

As described in the previous section, the slot-wise decoder largely generally treats each slot representation independently, except for predicting the important weighting scores for slots. Interaction and comparisons between the slots thus happens rarely, and only at the end in the reconstructed visual space and this can be limiting for learning.

Conversely, the set-based decoder approach takes the entire set of latent representations as input for its decoding process. In this way, the decoder can have a global view of the entire scene drawing information from all object-representations and allowing their latent representations to richly interact and compete directly in the latent space.

This set-based approach is similar to other deep learning methods and thus can

borrow directly from others that have been shown to perform well at scale.

In DINOSAUR, Seitzer et al. [240] the approach uses a plain Transformer decoder [272] to autoregressively decode each visual patch, conditioning on the set of object-representations from the encoder. This approaches uses a scalable architecture that has shown impressive results in many domains like the Generative Pre-trained Transformer (GPT) [221].

6.2 Method: Attentional Slot Decoder

In this section, we introduce the Attentional Slot Decoder, a method that uses a cross-attention mechanism in the latent space to allow for rich interaction between objects, while only requiring one decoder forward pass to reconstruct the final output. Our approach was inspired by the complementary advantages of the set-based and slot-based decoding methods, which we will now briefly discuss as the motivation for our design choices.

6.2.1 Attentional Slot Decoder

In this subsection, we present the Attentional Slot Decoder which has a very simple design that fulfils all the considerations above about using standard, scalable and resilient components.

Conceptually, our method utilise a cross attention layer from each decoding position to the set of object latents obtained from the encoder. We then form a grid of embeddings as a weighted average of the slots using the cross-attention score. This grid of embeddings now serves as the input for the scene decoder module to obtain the final scene reconstruction. Since the attention layers are permutation invariant, our decoders are also permutation invariant with respect to the object representation. In addition, since each vector in the latent set is treated as an independent representation, their compatibility are encouraged. Finally, iteratively forming the decoder's embeddings through the cross-attention mechanism allows for rich interactions and transformations between objects in the scenes, while the expensive step of decoding to the input space only needs to be executed once.

Figure 6.1 illustrates the conceptual similarities and differences between our proposed method and other approaches. Our method effectively combines the advantages and avoids the limitations of both slot-based and set-based object-decoding methods, as will be demonstrated throughout the following subsections.

We now summarise our method in Algorithm 1 and give a detailed description below.

Let us denote the set of K slot representations as $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_K] \in \mathbb{R}^{k \times d}$ that

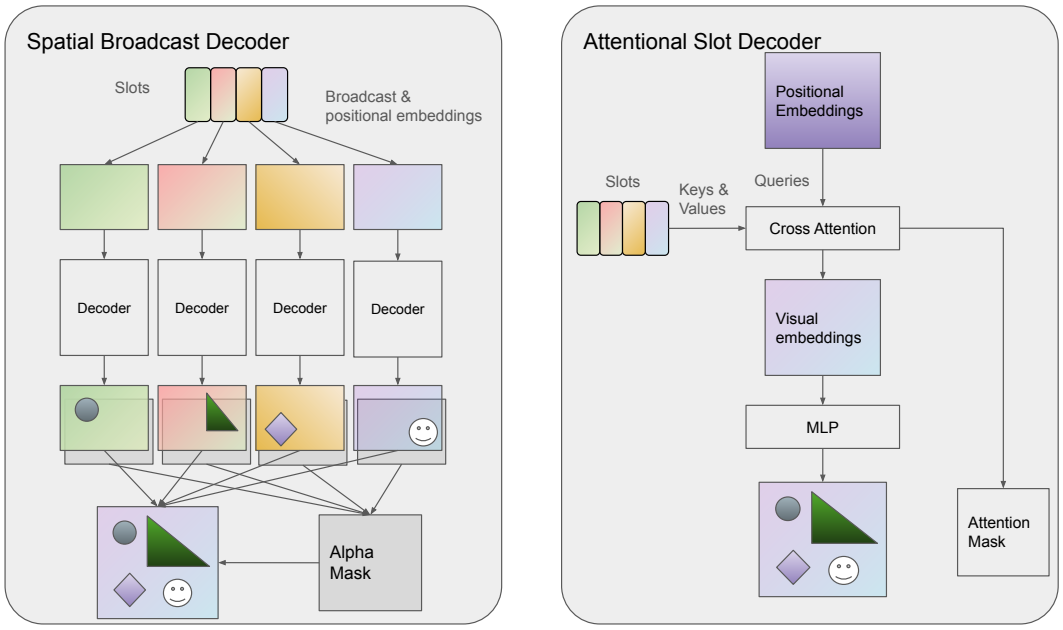

Figure 6.1: Overview of three different approaches to designing the decoder module for object-centric representation learning methods. A) Slot-based decoding. B) Set-based decoding method. C) The proposed Attentional slot decoding method.

we obtained from the output of the slot encoder. Each vector $\mathbf{s}_i \in \mathbf{S}$ would ideally capture the information of an object in a multi-object scene and the goal of the decoder would be reconstruct the original scene from the set of object representations.

We also denote $\mathbf{g} \in \mathbb{R}^d$ as the optional *global* scene representation. This global representation of the scene can be obtained by pooling from the grid of visual representations $\mathbf{V} \in \mathbb{R}^{h \times w \times d}$ or extracting from the special class-token of the Transformer architecture.

Decoder's queries: First we initialise a 2D spatial grid (3D spatial-temporal if working with video) with Fourier positional embeddings [266]. At this stage, the decoder queries consist only of positional information of the pixels (or patches) that it corresponds to. This is the same for all input samples.

To optionally inject scene-specific information, we could broadcast the global scene embedding $\mathbf{g}$ and add to every spatial dimension of $\mathbf{Q}$ as:

$$\mathbf{Q}_{x,y} = \text{PositionalEmbedding}(x, y) + \mathbf{g}; \tag{6.1}$$

for $x \in (0, h), y \in (0, w), \mathbf{Q} \in \mathbb{R}^{h \times w \times d}$. The spatial dimension of h and w can be as big as the original input or can be a down-scaled version to reduce the computational complexity, depending on the visual decoding module.

Algorithm 1 Slot Decoder Algorithm

Input:
$\mathbf{S} = [s_1, s_2, ..., s_K] \in \mathbb{R}^{k \times d}$ {Object Slots}
$\mathbf{g} \in \mathbb{R}^d$ {Global scene}
Initialise Decoder's Queries
$\mathbf{Q} \leftarrow \mathbf{P} \leftarrow \text{PositionalEmbedding}(h, w) \in \mathbb{R}^{h \times w \times d}$ {Positional queries}
$\mathbf{Q} \leftarrow \mathbf{Q} + \mathbf{g}$ {Positional and scene-specific queries}
Compute decoder embedding
for each of T iteration **do**
  $\mathbf{K} \leftarrow \text{LN}_\text{k}(\text{MLP}_\text{k}(\mathbf{S})) \in \mathbb{R}^{k \times d}$ {Key matrix}
  $\mathbf{V} \leftarrow \text{LN}_\text{v}(\text{MLP}_\text{v}(\mathbf{S})) \in \mathbb{R}^{k \times d}$ {Value matrix}
  $\mathbf{A} \leftarrow \text{softmax}(\mathbf{Q}\mathbf{K}^T) \in \mathbb{R}^{hw \times k}$ {Attention scores}
  $\mathbf{O} \leftarrow \mathbf{A}\mathbf{V}^T \in \mathbb{R}^{hw \times d}$ {Output matrix}
end for
Visual Decoding
$\mathbf{O} \leftarrow \mathbf{O} + \mathbf{P}$ {Add positional information}
$\mathbf{Y} \leftarrow \text{MLP}_\text{o}(\mathbf{O}) \in \mathbb{R}^{h \times w \times c}$ {Output}

Attention's keys and values: These are obtained from the set of object latents by using a linear layer followed by Layer Normalisation (LN) [10])

$$\mathbf{K} = \text{LN}(\mathbf{S}\mathbf{W}_K), \tag{6.2}$$

$$\mathbf{V} = \text{LN}(\mathbf{S}\mathbf{W}_V). \tag{6.3}$$

$\mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d}$ in Eq. 6.2 are the corresponding weight matrices for the linear layers.

Cross Attention operation: We now perform a standard cross attention operation with the set of queries, keys and values obtained from equations (6.1) and (6.2). The softmax normalisation operation is performed over the key dimensions of the attention matrix $\mathbf{A} \in \mathbb{R}^{hw \times k}$. The final output of the cross attention module is the weighted average of $\mathbf{V}$ based on the attention scores $\mathbf{A}$ $\mathbf{O} \in \mathbb{R}^{hw \times d}$.

$$\mathbf{A} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{6.4}$$

$$\mathbf{O} = \mathbf{A}\mathbf{V}^T. \tag{6.5}$$

These cross attention operations can be repeated over many iterations where the output $\mathbf{O}$ serves as the query.

6.2.2 Visual Decoder

With a spatial or a spatial-temporal grid of embeddings, we can pass these through a visual decoder module to obtain the final reconstruction in the output space. We follow standard practice and use a 1×1 convolutional network, effectively a pixel-wise MLP, to decode these embeddings into RGB values $\mathbf{O} \in \mathbb{R}^{hw \times 3}$.

$$\mathbf{Y} = \text{MLP}_O(\mathbf{O}) \tag{6.6}$$

Crucially, we add a residual connection from the positional embedding to the output of this attention mechanism. We find it necessary to inject positional information back into the decoder's embeddings in order for the model to be able to reconstruct and place objects in the correct location in the scenes.

The key insight to our approach is that we do not measure the contributions of object embeddings in the pixel space by decoding an alpha mask. We could directly let the object embeddings interact and compete to explain the *decoder's query* before it is decoded. The key-normalised attention scores now serve as the alpha masks, which states which object each pixel belongs to.

Due to disentangling the positional information and semantic information into the query and the key, the cross attention block is permutation invariant with respect to object embeddings. Each vector in the set is treated as an independent token and their compatibilities are encouraged due to the dot product similarity operation in computing the attention.

After constructing the decoder's query from object representations, our method is agnostic to the design of the visual decoder to the input space. We can use more powerful visual decoders for complex scenes and datasets, or use simpler, pixel-wise decoders for scenes with smaller objects.

6.3 Results

In this section we describe the experimental details and results in evaluating our proposed decoder method.

We demonstrate the effectiveness of our propose Attentional Slot Decoder on the Multi-Object Video (MOVi) dataset [97]. This is a synthetic multi-object dataset that was created specifically for the study and development of unsupervised multi-object video understanding. Being simulated and rendered programmatically, it provides not only high-quality and realistic videos but also provides rich and dense annotations for segmentation masks, depth, optical flow, surface normals and object coordinates. The annotations, while are not needed for the purpose of training unsupervised representation learning methods, are crucial for evaluation and are very

expensive to collect in the real life. We chose these two subsets, MOVi-A and MOVi-C to demonstrate that our decoder can work in both a simple environment similar to prior works, and also scale up to more challenging environments, representing an improvement for scaling object-centric methods to real-world datasets.

We follow the training and evaluation protocol of prior state-of-the-art works that use SAVI [150] for the most part, and focus on comparing our Attentional Slot Decoder with their counterparts. For a more detailed review of SAVi, please refer to Section 4 of this thesis.

Metric We use the Adjusted Rand Index (ARI) as the main metric for assessing the efficacy of video decomposition, object segmentation, and tracking. The ARI serves as a measure of clustering similarity, gauging the congruence between predicted segmentation masks and ground-truth masks in a manner that remains unaffected by permutations. This property of the evaluation metric makes it particularly suitable for evaluating unsupervised techniques. Similar to prior works [98, 170, 138], we compute the ARI for foreground objects, a version referred to as ARI-FG. In the context of video data, a singular cluster in the ARI calculation corresponds to the segmentation of an individual object over the entire video duration. This necessitates temporal coherence, with the absence of alterations in object identity, for achieving favourable outcomes on this metric.

6.3.1 RGB Reconstruction On MOVi-A

The Movi-A dataset, modeled after the popular multi-object CLEVR [136], is the simplest subset of MOVi and contains from 3 to 10 random geometric objects on a simple gray background. Despite it visual simplicity, it already posed major challenges for many earlier object-centric models [149, 98].

During the training phase, we divide each video into consecutive sub-sequences containing 6 frames each, where the initial frame receives the conditioning signal. We condition the signal with the bounding box information of each object in the first frame. On the MOVi-A dataset, we use 11 slots for the object latent representations. We train on videos with a resolution of 128x128 for 50,000 steps on a single GPU with 24GB memory, utilising a batch size of 16. The model is optimised using the Adam optimizer [148] with an initial learning rate set at 0.0002. This dataset contains a total of 10,000 videos, with 2,500 videos reserved for validation and the remaining 97,500 are used for training.

Qualitative Assessment: In Figure 6.2 we visualise some samples from the MOVi-A dataset to show our proposed decoder is capable of decoding to RGB pixel

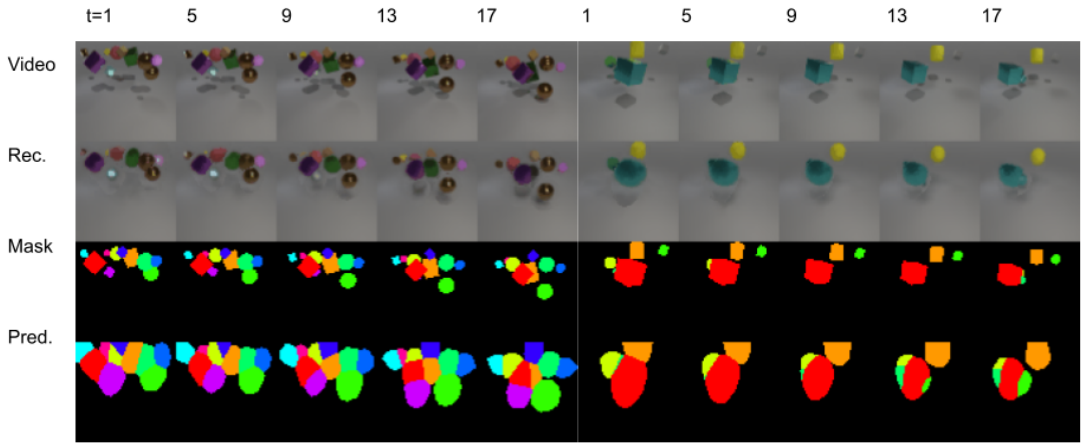

Figure 6.2: Some qualitative result of our Attentional Decoder on the Movi-A dataset. In each row of images we visualise the input video, the RGB reconstruction target of the input (Rec.), the ground truth object masks (Mask) and the predicted object masks from our object representations (Pred.). On the left we show examples with many objects in various shape and colours. On the right, we show simpler examples. Notably, our model fails to capture all the objects (shown with the green segmentation mask in the top right corner).

values. The samples on the left are for a scene with many small objects, with complicated interactions, yet our method can still reconstruct the original video and is able to segment most of the objects without supervision based on the object-centric representation. The second sample on the right visualises a representative failure case, where one small object completely fails to be reconstructed or segmented.

In general, we observed that the object segment masks tend to be inflated in size compared to the original object masks. The inflated regions upon inspection, usually expand out to cover the shadows of the objects. This inclusion of shadows in the object masks are consistent over many timesteps in the video.

Quantitative Comparison: In Figure 6.3, we compare the efficiency gain by our method compared to the baseline Spatial Broadcast Decoder [283] used by SAVi. This shows that during training, our proposed decoder learns faster than the baseline and in the end achieves slightly better performance on the ARI-FG metric. Details are listed in Table 6.1.

In addition to the positional embeddings used as the initial queries for the cross attention module, we found that it is also important to provide the visual decoder with positional information as well. Without that, the performance on the ARI-FG

Figure 6.3: The ARI-FG metric evaluated on the train and validation sets during training of the base line SAVi decoder and our Attentional Slot Decoder on MOVi-A dataset. This shows that our proposed method is able to learn to segment objects in an unsupervised manner faster and with more stability than the baseline.

Table 6.1: Comparison between the baseline SAVi and our proposed method. While we achieve similar performance on ARI-FG on the training set, we achieve slightly better performance on the validation set while requiring 4 times less memory to train. Ablation results of our method without the positional embedding added before visual decoding and without the global scene embedding are also provided.

Method	**ARI-FG train**	**ARI-FG val**	**Memory (GB)**
SAVi	0.9115	0.8389	24
Our Method	0.9225	0.8488	6
Without - Pos. Emb.	0.7814	0.5933	6
Without - Global scene.	0.7891	0.6279	6

metric drops by 0.13 in absolute score on the validation set, as indicated on line 3 of Table 6.1.

Similarly, without adding the global scene representation to the initial queries,

we saw a performance drop in the object discovery metric. From here on, we use our method with added positional information and global scene representation as the default.

6.3.2 Optical Flow Reconstruction On MOVi-C

The MOVi-C dataset is a substantial step up in visual complexity compared to the MOVi-A dataset. It replaces simple geometric objects in MOVi-A with complex, everyday objects scanned in the real world. The backgrounds and lighting are randomly selected from a set of HDR images, which makes this variant substantially more challenging than MOVi-A to learn.

Object-centric methods tends to rely on low-level visual cues to segment objects [142]. Due to the increase in visual complexity, this dataset poses a major challenge for many object-centric learning methods when using RGB reconstruction as the training target. Following the baseline, and also to show the versatility of our method on different domains, we use the optical flow reconstruction task to evaluate the performance of our method. Apart from that we use the same setup as in the experiment with MOVi-A.

Qualitative assessment: In Figure 6.4, we visualise some samples from the MOVi-C dataset. The first sample on the left demonstrates the model's capability to accurately reconstruct optical flow signals with many independently moving objects. The sample on the left shows a failure case where the model fails to accurately reconstruct less regular objects.

Due to using the optical flow as training signal, the model also tends to group visually-separated objects with the same movement in the same object representation (top left corner).

Quantitative comparison: Similar to the MOVi-A dataset, we compare the training efficacy of our method to the baseline in Figure 6.5 and in Table 6.2.

Table 6.2: Comparison between the baseline SAVi and our proposed method. While we achieve similar performance on ARI-FG on the training set, we achieve slightly better performance on the validation set while requiring 4 times less memory to train.

Method	**ARI-FG train**	**ARI-FG val**	**GPU Memory (GB)**
SAVi	0.8155	0.6053	24
Ours	0.8425	0.6557	6

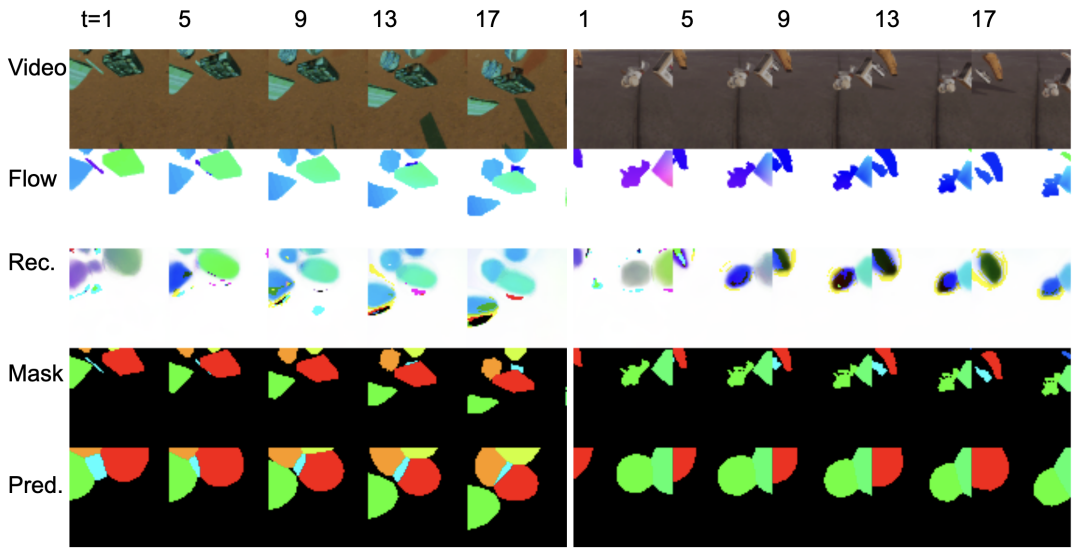

Figure 6.4: Some qualitative results of our Attentional Decoder on the MOVi-C dataset. In each row we visualise the input video, the optical flow, the RGB reconstruction target of the input (Rec.), the ground truth object masks (Mask) and the predicted object masks (Pred.) from our object representations. On the left, we show an example with many objects of various shapes and colours. On the right, we show a simpler example showing our model failing to capture all the objects (with green segmentation mask on the top right corner).

Once again, even on a more challenging dataset and with a different reconstruction modality, our simple method yields a 5% increase in the ARI-FG score while requiring significantly less memory and compute time compared to the SAVi baseline.

6.4 Discussion

In this chapter we have proposed a simple Attentional Slot Decoder for object-centric representation learning methods in the framework of autoencoding. Our simple approach combines the strength of both slot-based and set-based decoding, and thus made progress toward the question of improving the efficiency, as stated in our Research Question 4. The core idea of our method is the utilisation of a cross attention module between the positional decoder query with the object representations. This rich interaction in the latent space allows for the exchange of semantic object information.

The decoupling between measuring object interaction and decoding at each position allows the expansive visual decoding component to be run only once for each sample. This results in a tremendous amount of saving in memory and compute time needed to train such object-centric learning models. This saving will scale up

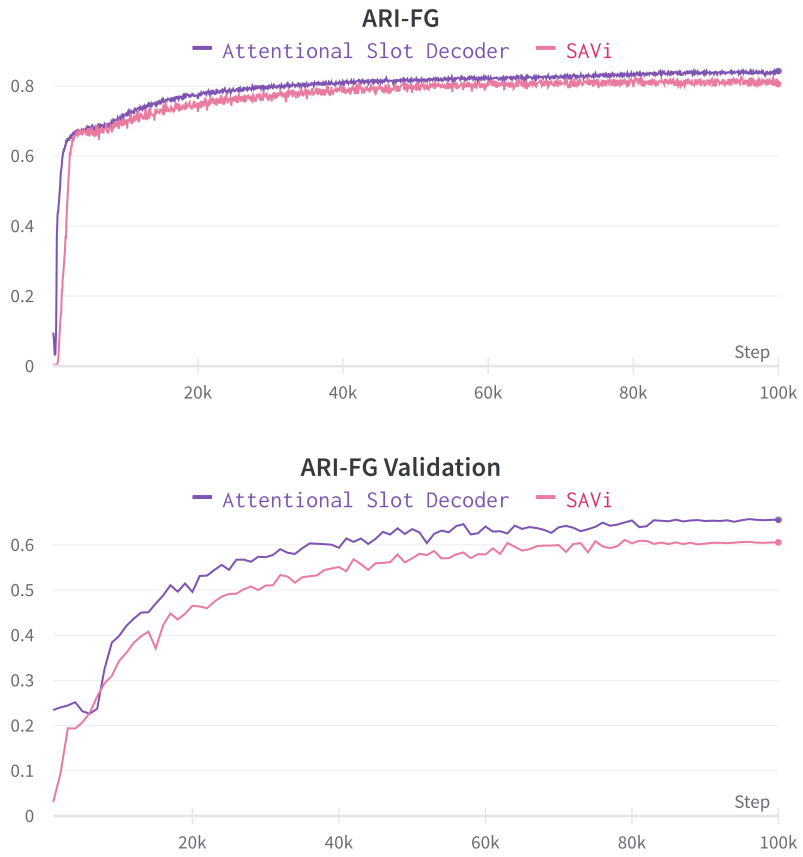

Figure 6.5: The ARI-FG metric evaluated on the training and validation sets during training of the base line SAVi decoder and our Attentional Slot Decoder on the MOVi-C dataset. This shows that our proposed methods are able to learn to segment objects without supervision faster than the baseline.

further in more complicated datasets with even more objects.

All else being equal, our proposed decoder learns faster than the baseline, achieves comparable performance while requiring substantially less memory and compute time.

Concurrent to our work, a similar idea has been proposed in [234] in a different setting of multi-image novel view synthesis. This further validates the motivation and applicability of our approach.

In this work, we mostly focus on the decoupling between forming a decoder query via latent object interaction. For the visual decoding, we broadly follow prior work and use a pixel-wise decoder. Future work could further investigate the benefit of this approach and expand it to use even more expressive and powerful visual decoding for more challenging scenes and datasets. Another interesting direction would be to combine this with other decoding approaches such as Masked AutoEncoding [109] for further benefit.

Chapter 7

Object Discovery with Geometric Representation

In recent years, the fields of learning visual representation and Deep Learning have seen remarkable progress. These advancements have often been driven by scalable architectures like the Vision Transformer (ViT) [67], efficient hardware utilisation [109], and the availability of large and diverse datasets [249], or a combination of these elements [207].

Object-centric representation learning has progressed significantly in recent years as well, evolving from proof-of-concept methods in 2D block-world scenarios [28, 98, 149] to handling more intricately rendered 3D environments [170]. However, it still faces challenges when applied to more realistic and complex datasets, as highlighted by Kipf et al. [150]. In particular with the pre-training task of image autoencoding, object-centric models tend to rely heavily on RGB colour values for both the reconstruction and object-discovery tasks, limiting their scalability to more complex visual scenes. Exactly how to apply these lessons from large-scale visual representation learning, or leverage their capabilities, to scale up object-centric representation learning methods is an open and exciting challenge.

In Chapter 6 and Chapter 5, we focused on different inductive biases for designing network architectures to facilitate the learning of object-centric representations. Concerning both Research Question 3 on advancing the object-centric method on more complex scenes, as well as Research Question 4 on improving its efficiency, in this chapter, we shift our focus to an equally crucial aspect of any learning method: the data and learning signals it relies on. Motivated by the advances in pre-training visual representations, we have decoupled the challenge of learning *object* representations from that of learning *visual* representations, capitalising on the parallel progress of each.

In particular, our attention is directed towards a specific category of visual representation that incorporates a richer set of geometric information. This approach is

aimed at enhancing both object discovery and learning hierarchical representations, and further addresses Research Questions 3 and 4.

The primary contribution of this chapter lies in our exploration and demonstration of effective object-centric learning techniques, along with an investigation into the role of depth information in learning object-centric representations. Developing a means to acquire object-centric representations without the need for explicit and costly annotations more closely aligns with the way humans naturally learn, representing a significant step toward robust representation learning.

7.1 Related Work

In this section, we briefly review progress in two different areas, namely large scale visual representation pre-training and object discovery beyond RGB reconstruction. This will provide the context for our attempt to combine the advantages of both approaches.

7.1.1 Pre-trained Visual Representations

There has been a trend in the field of large-scale pre-training in recent years, where different types of data, including text, images, video, and audio, can now be used with a standardised and scalable architecture called the Transformer [272]. The Transformer architecture, in its various forms — original encoder-decoder, encoder-only (e.g., Bidirectional Encoder Representations from Transformers (BERT)) [61], and decoder-only (e.g., GPT [221]) —employs a self-attention mechanism to process a set of tokens, where each token represents a piece of input information. These sets of tokens are processed via the self-attention mechanism [14], which queries the most relevant information to each token.

In the case of visual representation learning, a Vision Transformer model initially yields tokens based on non-overlapping patches of the images. Along with additional positional information, these tokens are then iteratively refined by interacting with other patches across the entire image, forming what can be seen as feature maps that retain spatial (and temporal) information similar to a Convolutional Neural Network (CNN).

The output of this network is subsequently pooled to create a final global representation of the scene. In some instances, a special token called 'CLS' is concatenated with the initial set of patch tokens and is jointly optimised via the self-attention mechanism throughout the network's layers. Please refer to 2.3 for a more detailed discussion of architecture.

By scaling the pre-training of these models with more parameters, on more data,

and with more computational resources, the learned representations have proven to be widely useful and achieve competitive or state-of-the-art performance across a wide range of downstream tasks.

Scaling supervised model Recently, Dehghani et al. [57] have introduced a method for scaling Vision Transformer (ViT) models up to a staggering 22 billion parameters on a weakly-supervised classification task. When trained on a dataset of over 4 billion images, the results of this scaled model highlight the advantages of such scaling, including a more favourable performance-fairness trade-off and better alignment with the human visual system in terms of shape and texture biases.

Self-supervised discriminative method Beyond supervised or weakly-supervised pre-training, there are also noteworthy developments in the area of self-supervised pre-training tasks. One such method is Self-**di**stillation with **no** label (DINO) [33], which employs a discriminative self-supervised pre-training approach through a self-distillation mechanism. DINO can be viewed as a form of contrastive learning but without explicit negative pairs. Notably, it has demonstrated intriguing emergent properties, including object-segmentation masks, which arise from purely discriminative pre-training when paired with "global-to-local" correspondence, as discussed earlier in Section 3.2.2. More recently, DINOv2 [207] extends the original DINO implementation with better losses and regularisation methods from Image BERT Pre-Training with Online Tokenizer (iBOT) [313] and Swapping Assignments between multiple Views of the same image (SwAV) [34] on more curated datasets.

Autoencoding method Another interesting direction is the Masked-AutoEncoder (MAE) [109] that uses generative and reconstruction tasks during pre-training. A recent paper entitled "Masked Autoencoders Are Scalable Vision Learners" presents a novel MAE approach that leverages transformers and autoencoders for self-supervised pre-training and outperforms fully-supervised approaches on some tasks. MAE-based methods [109, 84, 249, 241] work by randomly masking a substantial portion of the image patches which can be up to 90% of the image before being processed by a large Vision Transformer encoder. Thanks to the independence and non-overlapping nature of patches in Vision Transformer (ViT), and the positional information in the form of extra embeddings, the Transformer encoder can process only the non-masked input patches, which in turn enables scaling to larger models on the same hardware and compute budget. To learn effective representations, the representation of the unmasked patches is then concatenated with special mask tokens. A smaller and lightweight Transformer then jointly attends to all the tokens, visible and masked, to decode the original patches at the masked positions. This clever

asymmetric design in encoder and decoder enables high hardware utilisation while also not wasting much capacity on the decoder component, which is not the goal of representation learning.

7.1.2 Object Discovery with Geometric Priors

Scaling object-centric methods for large-scale, real-world, and complex datasets has proven to be a formidable challenge. As discussed in Chapter 4, most existing approaches are rooted in the autoencoding of visual inputs. While this approach works effectively on visually simple datasets with a limited range of colour variations, it struggles when applied to more realistic datasets that demand a deeper semantic understanding.

In contrast to visual representation learning, where reconstructing visual input, or predicting masked input can encourage the learning of useful representations, such a strategy may fall short for object-centric learning. This is because object-centric learning aims not only to learn but also to segregate information into different object slots, while autoencoding methods tends to learn global representation of the scene due to its bottleneck structure in the latent representation [30].

By training to reconstruct RGB input values, these systems can inadvertently learn to group input signals with similar colours and texture to the same objects. While this approach is sufficient for simple synthetic scenes, it has been shown to fail on real-world datasets, or even synthetic datasets with more challenging textures.

Incorporating alternative training signals beyond the mere reconstruction of RGB values, such as optical flow and depth information, holds significant potential for enhancing the scalability and performance of object-centric learning.

Recent work, specifically SAVI by Kipf et al. [150], has introduced an innovative approach using optical flow as a training signal. This approach excels on more complex datasets, capitalising on the consistent movement of pixels belonging to the same object to group input signals. While it outperforms traditional methods that rely solely on reconstructing RGB pixels, it is somewhat sensitive to changes in viewpoint due to its reliance on low-level optical flow features.

Building upon the foundation laid by SAVi, SAVi++ [71] takes this strategy a step further by employing sparse depth information as the training signal. This not only enhances performance beyond that of optical flow but also exhibits greater robustness to changes in viewpoint. The persistence of depth differences between objects remains intact even when the viewpoint shifts. Notably, SAVi++ has achieved scalability in object-centric methods when applied to the real-world Wayve self-driving dataset [263].

It is important to note that both of these approaches require additional paired

signals, such as RGB input images and optical flow or depth information. While this requirement can be somewhat mitigated by using the outputs of some pre-trained optical flow and depth prediction modules, it still results in object-centric learning systems that depend on extra supervisory signals, either directly or indirectly.

An exciting avenue for future research lies in exploring how to combine the strengths of these approaches to achieve more robust and scalable object-centric representations.

7.2 Methods

In this section, we present our approach to unsupervised object discovery and object-centric representation learning. We call this CrObject: Cross-view completion pre-training for Object-centric learning. It is based on a pre-trained visual representation, as illustrated in Figure 7.1. The method follows the general autoencoding pipeline, but instead of predicting raw visual input as RGB values, we leverage a pre-trained vision model as the backbone to learn object-centric representations on top of it. To address more challenging settings and datasets, we modify the reconstruction target to focus on features extracted from the frozen visual backbone.

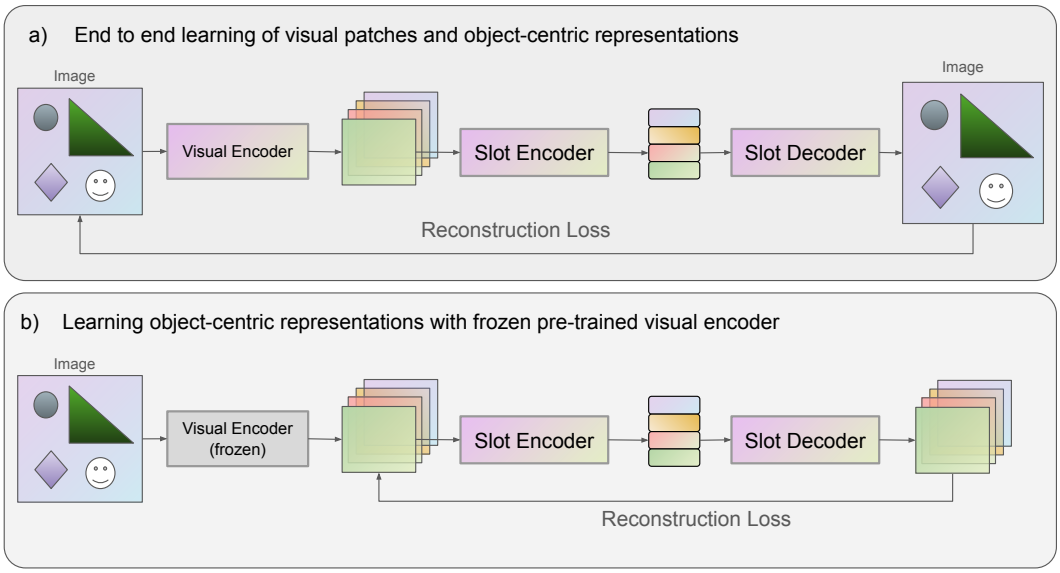

Figure 7.1: Overview of our proposed method CrObject. Input images or video are encoded into visual tokens by a pre-trained CroCo model. A Slot Attention module then parses them into a set of object's slots. From this, the Attentional Slot Decoder reconstructs the original feature maps of CroCo.

Importantly, the selection of the backbone is guided by the motivation to incorporate geometric signals that assist in object discovery. Overall, our pipeline is char-

acterised by its simplicity, efficiency, and its contribution to scaling object-centric learning towards more challenging and realistic settings while remaining entirely end-to-end and self-supervised.

We first describe the architecture of our pipeline that enables learning object-centric representations decoupled from visual representations. Then we will describe the criteria for choosing the pre-trained model that helps with object-centric learning.

7.2.1 Architecture for Decoupling Visual and Object Representation Learning

Visual encoder: The input images or videos $\mathbf{X} \in \mathbb{R}^{h \times w \times 3}$ are first encoded via a visual backbone. To tackle more challenging scenes, the visual encoder must be sufficiently expressive to capture and represent the differences in texture and lightning for different objects. To this end, we use a standard Vision Transformer as the backbone due to its state-of-the-art performance and the availability. These visual embeddings $\mathbf{V} = \text{ViT}(\mathbf{X}) \in \mathbb{R}^{h' \times w' \times d}$ are downscaled by a factor of patch size p from the chosen ViT model, usually equal to 16 $h' = h/p, w' = w/p$.

Object encoder: Following previous works and similar to Chapters 6 and 5, we use a Slot Attention module to further obtain a set of object representations from the visual representations: $\mathbf{O} = \text{SlotAttention}(V) \in \mathbb{R}^{k \times d}$

Object decoder: An object decoder is needed to convert the set of object features to a feature map. For this, we utilise the Attentional Slot Decoder introduced in Chapter 6: $\mathbf{Y} = \text{AttentionalSlotDecoder}(\mathbf{x}) \in \mathbb{R}^{h \times w \times d}$. Here, instead of predicting the input pixels $\mathbf{X}$, we instead predict the feature maps obtained from the visual encoder $\mathbf{V}$.

Visual decoder: If the pre-trained visual backbone has a corresponding decoder, we can reconstruct the original input from the predicted representations $\hat{X} = \text{Decoder}(\mathbf{Y})$. During training, however, this module is omitted to save memory and compute resources since it is not needed to compute the loss.

Training objective: We can now compute the reconstruction loss on the *feature space* from our pre-trained visual encoder. Here we use a simple Mean Squared Error on the feature space: $\mathbb{L} = \|\mathbf{Y} - \mathbf{V}\|_2^2$.

7.2.2 Geometric Prior from Self-Supervised Features

A crucial difference in our approach is the use of a pre-trained vision model to build object representations on top of, and also to serve as the target of object reconstruction. Given the proliferation of pre-trained vision models in recent years, the question arises: which pre-trained model should be employed for object-centric learning, and what motivates this choice?

Emergence of "object" in biological intelligence Across the spectrum of biological intelligence, from adult humans to infants and animals, there exists a seemingly inherent capacity to comprehend and interact with objects in their environment. Extensive research in cognitive development of children has shed light on the early acquisition of object permanence, a crucial cognitive milestone that becomes evident in children as young as five months of age [215].

One fundamental component underpinning the cognitive process of object perception is the stereoscopic input from human eyes. This binocular vision system is a strong cognitive bias by providing a strong signal to comprehend and represent our world in three dimensions. Associating every point with a spatial depth distance is the most powerful and obvious explanatory factor that explains this integration from the visual input of the left and the right eye. Furthermore, it effectively accounts for the parallax effect triggered by changes in viewpoint, whether through head or bodily movement in space.

Once the existence of objects is firmly established in the cognitive repertoire, this naturally implies the awareness that objects do not spontaneously come into existence, vanish or morph in appearance and property; instead, they transition through space and time with fluidity.

After encountering many different objects, humans exhibit an ability to classify and categorise them seemingly automatically. These classifications are largely contingent on the objects' appearance and behaviour, enabling broad categorisations.

Over time, more complex concepts about a scene with multi-objects and their interactions can be acquired: i.e. one object can cover or mask another object, or they can collide and alter trajectory. All other higher semantic meanings of objects from intuitive laws of physics to causal relationships, are all therefore built on top of the lower-level notion of objects.

Biological intelligence effortlessly builds a hierarchical representation of the world, with the foundation building blocks of depth and 3D geometric information. Inspired by this hierarchy of representation acquired by human intelligence, how can we emulate this for the task of object-centric representation learning? How does the current paradigm of pre-training vision models fit into this order?

Emergence of objects from semantics The initial breakthrough in self-supervised learning, surpassing supervised learning in vision tasks, can be attributed to contrastive learning methods (see Chapter 3). These models harness the similarities between pairs of input images to constrain their representations to be alike in the latent space. To counteract the issue of the network mapping all inputs to a single latent vector, various techniques have been developed, including the introduction of projection heads, momentum encoders, stop gradient operations, whitening procedures, and others.

Among these discriminative approaches, Caron et al. [33] shows that a rough saliency map of objects in a scene can emerge from purely self-supervised learning in the form of attention scores of the vision transformer model, as described in Figure 7.2. The pre-training objective of local-to-global correspondence is attributed to this emergence of object semantics.

On the other hand, autoencoding methods learn representations by reconstructing the input data without imposing explicit constraints on the latent representation other than to bottleneck the layer's dimensionality. Notably, the Masked Autoencoder (MAE) has demonstrated robust performance and efficiency, achieving impressive results in downstream tasks. By constraining the model to learn to reconstruct masked patches from non-masked patches, the model is encouraged to learn a global representation for any given visible patches [30]. This leads to a representation space that is very powerful for fine-tuning on many downstream tasks that rely on semantic information.

Attempting to combine the best of these two approaches, Masked Siamese Networks (MSN) [9] introduces the concept of mask-denoising without requiring pixel or token-level reconstruction. It operates by presenting two different views of an image, wherein MSN randomly masks patches in one view while keeping the other view intact. The objective is to train a neural network encoder, typically implemented with a Vision Transformer (ViT), to generate similar embeddings for both views. It implicitly performs the denoising operation at the representation level by ensuring that the representation of the masked input closely matches that of the unmasked input. This approach encourages the network to learn meaningful representations while avoiding the complexity of pixel-level reconstruction.

Overall, these approaches typically encourage a representation space with strong semantic information. When fine-tuned, these yield good results on downstream tasks such as classification and segmentation.

In the sub-topic of object-centric representation learning, most closely related to our work is DINOSAUR [240] which attempts to discover and learn object-centric representations with the semantic representation obtained from the likes of DINO [33] or MAE [109].

Emergence of objects from geometry Recently, CroCo [286] extended the MAE framework from the single image domain to the multi-view domain. Instead of decoding the masked patches from the unmasked patches of the same images, CroCo predicts the entire image of the scene when provided with the information encoded from another viewpoint of the same scene. In this way similar to contrastive methods, it utlises the similarity between images of the same scene from different viewpoints to organise its latent space. This cross-view completion mechanism endows the network with the ability to process stereo input, and similar to the development of human cognition, this leads to a representation space with strong geometric information.

This cross-completion task encourages the model to not only learn important semantic information in the scene but also to model the 3D geometric information in its representation.

In this chapter, we argue that to better mimic the hierarchy of representations in the human and animal brain, and simultaneously take advantage of large pre-trained visual models, we should focus on building object-centric representations on top of the visual model with strong geometric signals, first.

Our analysis and experimental results in the next section will provide evidence supporting this direction.

7.3 Results

In this section we first perform an analysis on the correlation between emergence of objects in network attention maps. Insights discovered here then lead us to present our experiment in enhancing object-centric discovery with geometric-aware visual representations.

7.3.1 Attention Maps Mostly Indicate Semantic Representations

In Figure 7.2, we visualise the attention maps from the last layer of the encoder, generated using various pre-trained visual models across different frames of a multi-object video sourced from the MOVi-C dataset [97]. Our analysis of four distinct pre-trained models reveals discernible behaviours, which we can classify into two distinct groups. The first group comprises DINO and MSN, both of which exhibit a pronounced focus on the objects within the scene. Given our scene's composition with numerous small objects, the attention maps of DINO and MSN collectively concentrate on what could be described as the foreground.

Conversely, the second group, encompassing MAE and CroCo, do not display any

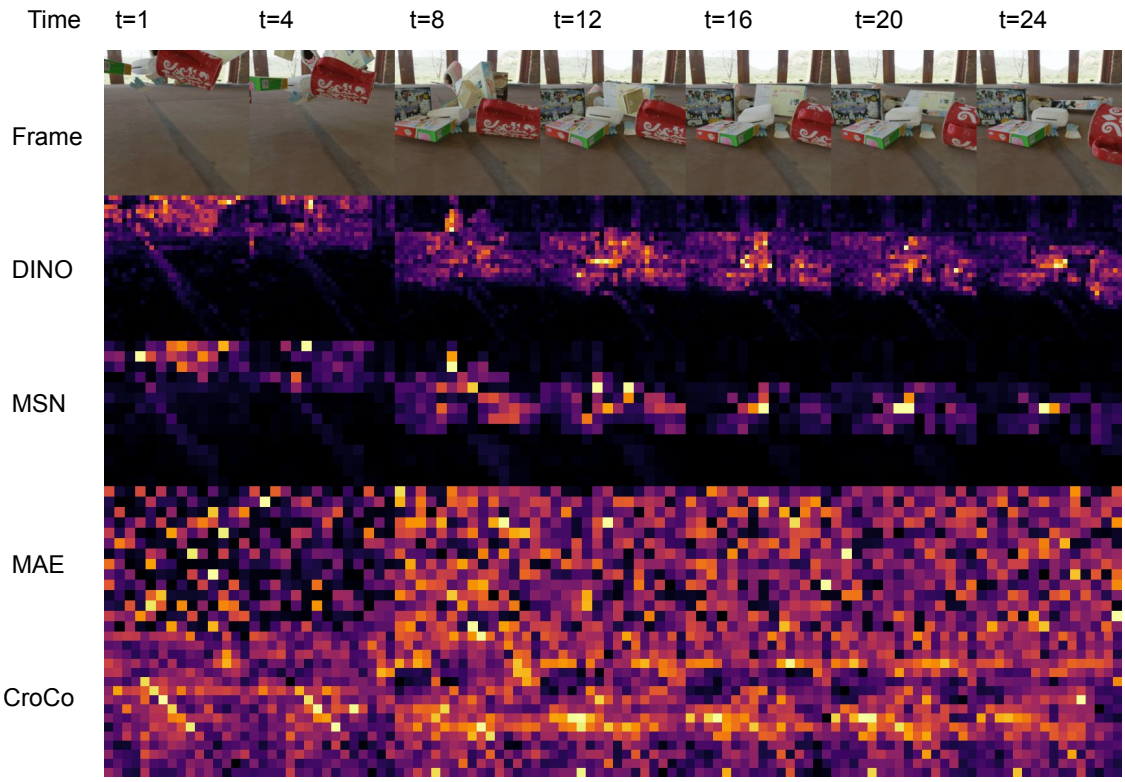

Figure 7.2: Attention map of different pre-trained self-supervised vision models on a video from the Movi-C dataset. While DINO and MSN show localised attention towards foreground objects, MAE and CroCo exhibit a diffused, global attention map.

conspicuous grouping behavior. Their attention scores disperse across the feature maps, extending even to areas corresponding to background elements in the original input images. Notably, despite this apparent "bug" in their attention maps, these models demonstrate strong performance after fine-tuning, remaining competitive with models from the first group.

One possible explanation for this divergence in attention style lies in their pre-training paradigms. While MSN and DINO employ self-supervised discriminative methods, MAE and CroCo rely on unsupervised generative models. The first group can trace their lineage back to earlier contrastive methods, albeit without the use of negative pairs, while the second group adopts an autoencoding framework, incorporating input-masking techniques.

Since the seminal work of DINO by Caron et al. [33], which demonstrated the emergence of objects in the attention maps of self-supervised training models, numerous subsequent works have capitalised on this insight to incorporate "objectness" into their own models, leveraging the representations inspired by DINO. Notable examples include the work in Kerr et al. [144] and Siméoni et al. [244].

This observation might lead one to assume that for object discovery and learn-

ing, methods like DINO or MSN should be preferable. Unintuitively however, our experiments, as shown in the next section, demonstrate that this is not the case. It becomes evident that lower-level information, not fully represented in the attention map, can significantly influence results on the task of unsupervised object discovery and learning.

7.3.2 Geometric Representation Improves Object Discovery

Task: In line with Chapters 6 and 5, we continue our emphasis on the objective of unsupervised object discovery, which was introduced in Chapter 4. In summary, our objective remains to acquire a collection of object representations from a given scene, where each slot in this collection corresponds to a high-level object within the scene. At present, a significant challenge lies in the automatic discovery of these independent objects within a scene with minimal supervision, a feat accomplished effortlessly by humans and animals.

Evaluation: At every decoded location, we assess the influence of each object slot on the output using its corresponding attention score. To establish the ground truth for prediction segmentation, we employ the *argmax* operator across slots.

Given the permutation invariance of slot representations, where each slot can potentially associate with any object in the scene, we employ the ARI-FG metric. This metric quantifies the similarity between the predicted segmentation mask and the ground truth mask, while disregarding the foreground class and the order of the remaining objects.

Baseline: We compare our approach with three other baselines that are highly relevant to our work. Firstly, we assess the original Slot Attention method, which introduced the widely-used Slot Attention module. Next, we evaluate DINOSAUR, a closely related method to ours, as it also utilises a pre-trained vision model for target prediction. Finally, we consider SAVi++, a method that scales up the SAVi architecture and incorporates explicit paired depth information as the target. For a more comprehensive overview of these baselines, please refer to Chapter 4.

Method: Our primary method aligns with the description provided in Section 7.2.1. We employ the pre-trained CroCo model [286] as both the backbone for the Slot Encoder module and the prediction target for the Slot Decoder module.

For consistency across all methods, we adhere as closely as possible to the training and evaluation protocol outlined in SAVi [150]. This includes training on video sequences, with the bounding boxes of objects in the first frame serving as the conditioning signal for initialising the slot representations.

Qualitative analysis In Figure 7.3 and Figure 7.4, we present the segmentation results obtained through our method on two samples extracted from the MOVi-E dataset. For each sample, the first row displays different frames from the video clip, with the ground truth segmentation tightly fitted on top. In the second and third rows, we showcase our predicted segmentation, either from the encoder or decoder attention map, superimposed on the frames for visual comparison.

In Figure 7.3, we have chosen two samples with a relatively stable camera, featuring several static objects and one or two moving objects. Our method demonstrates its ability to accurately segment most of the larger objects, such as the teddy bear, box, and shoe. However, the smaller objects along the left edge tend to be grouped together within the same object slot.

It is worth noting that since we carry over the object slots from the previous frame as the initialisation for the next frame, errors can accumulate over time, leading to a qualitative decline in segmentation (left to right). In the first sample, a portion of the background becomes erroneously segmented as an object as time progresses. Similarly, in the second sample, the bottom shoe is incorrectly segmented into two objects in later frames.

In Figure 7.4, we have chosen clips that present more challenging scenarios, featuring numerous smaller objects in motion and interacting simultaneously, all compounded by camera motion (often better observed in video format).

A notable trend in these cases is that smaller objects tend to be represented by the same object slot, as evidenced by the bottom border of the first sample. On the other hand, larger objects can sometimes be split between two different slots, as can be seen with the white pill bottle with the red label in the second sample.

In summary, these challenges highlight significant opportunities for improvement, both in terms of visual resolution and the temporal consistency of object slots.

Quantitative analysis In Table 7.1 we compare quantitatively, our result on the object discovery task against the baselines.

As hypothesised earlier, we observe a gradual improvement in performance as more training signals are included in the prediction target. Across both the MOVi-C and MOVi-E datasets, we surpass the performance of both SAVi and DINOSAUR, the two methods most closely related to our proposed approach.

DINOSAUR [240] relies on the *semantic* representations of DINO [33] as the prediction target, in contrast to our approach, which utilises the *geometric* representations of CroCo [286]. While both methods fundamentally use a pre-trained model to initiate the object discovery and learning process, it is worth highlighting that our method achieves a significant improvement of 10%. In contrast, Seitzer et al. [240] reported no noticeable difference when substituting different pre-trained

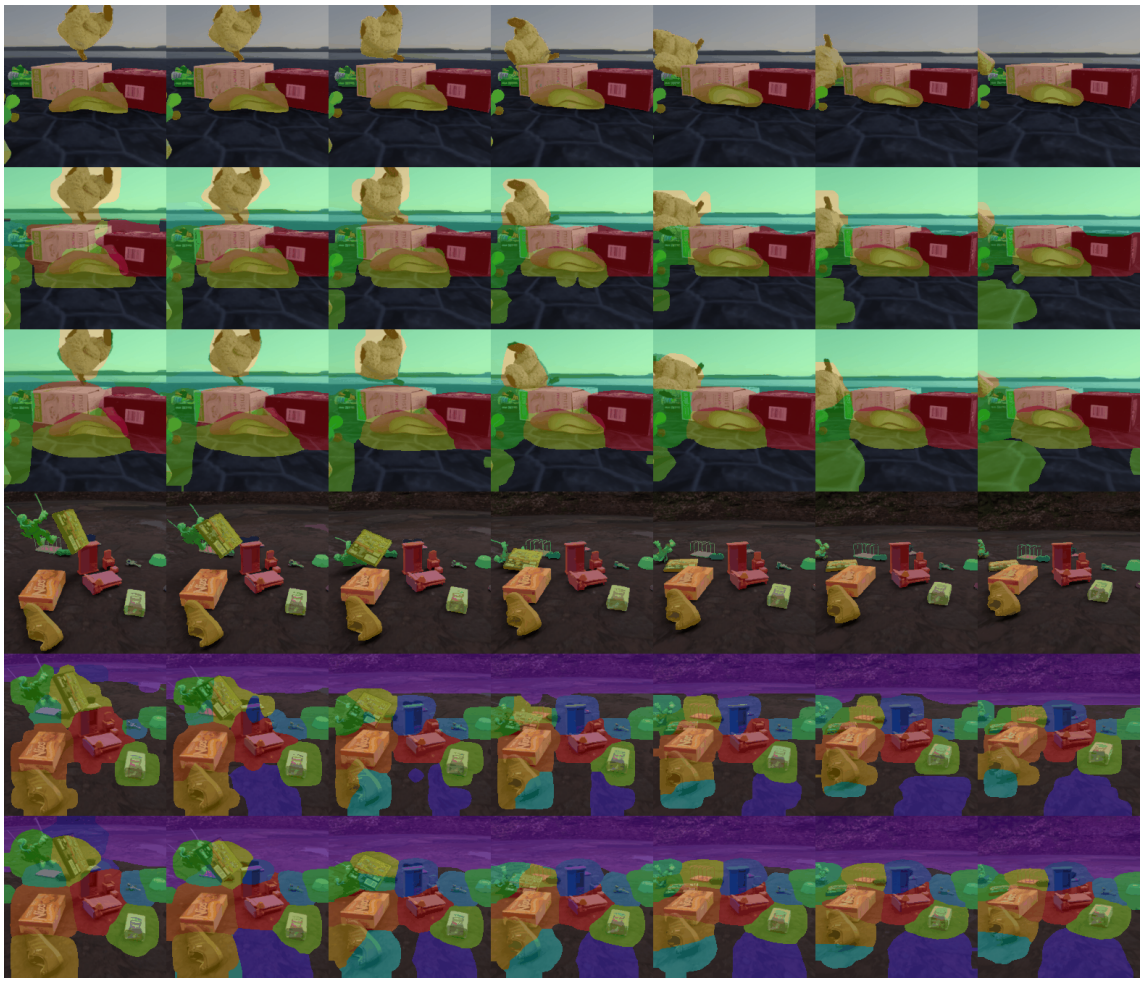

Figure 7.3: Two examples of unsupervised segmentation of objects with our method on the MOVi-E dataset. The horizontal axis represents different timeframes in a clip while the vertical axis shows our prediction. The first row shows input images with ground truth masks overlaid, the second row is overlaid by the segmentation from our slot attention encoder and the third row overlaid by our prediction from the attentional slot decoder.

Table 7.1: Comparison between the performance of our method and the baseline on the MOVi-C and MOVi-E datasets using the ARI-FG metric with values from 0 to 1 (higher is better). We also list the prediction target of each methods as an explanation for the performance differences.

Method	**MOVi-C**	**MOVi-E**	Prediction Target
SAVi	0.438	0.450	RGB pixels
DINOSAUR	0.686	0.651	Semantic features
Ours	0.788	0.766	Geometric features
SAVi++	0.8425	0.823	Depth values

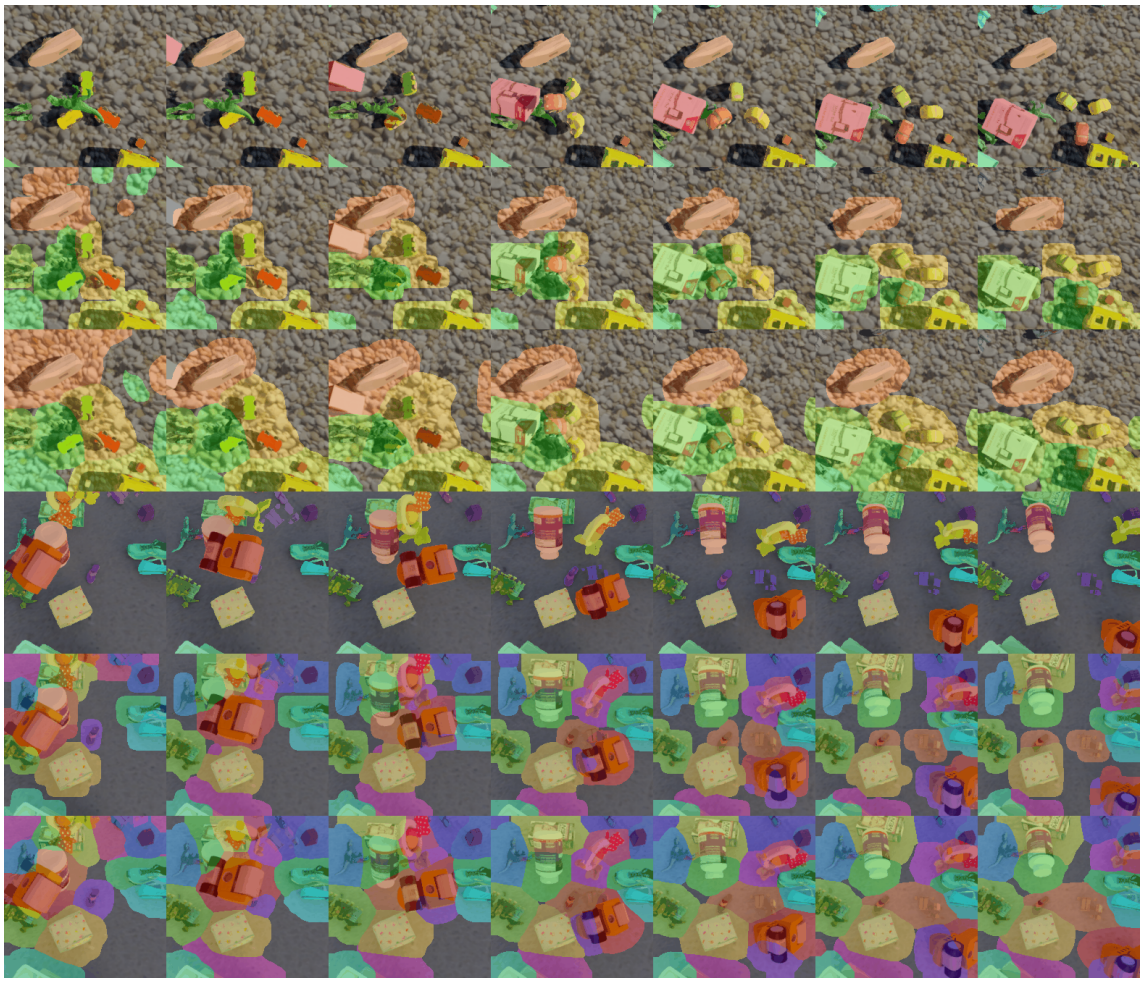

Figure 7.4: Two examples of unsupervised segmentation of objects with our method on the MOVi-E dataset. The horizontal axis represent different timeframes in a clip while the in the vertical axis shows our prediction. The first row shows input images with ground truth masks overlaid, the second row is overlaid by the segmentation from our slot attention encoder and the third row overlaid by our prediction from the attentional slot decoder.

models like MAE or MSN in place of DINO.

We attribute this improvement to the distinctive characteristics of *geometric* and *semantic* representations, as elaborated earlier. This is further supported by our comparison with SAVi++ [71], which explicitly utilises depth values as a training target.

Our method marks a significant step towards the goal of unsupervised discovery without the need for additional supervision signals, such as depth or optical flow. This is accomplished through the strategic selection of a self-supervised vision backbone, thus successfully addressing both Research Questions 3 and 4 on improving the efficiency and capability of object-centric learning methods.

7.4 Discussion

In this chapter, our focus shifted from architecture biases, as discussed in the previous chapter, to the learning signals used for object-centric representation. By harnessing a pre-trained vision model enriched with geometric information in its latent space, we established a comprehensive pipeline for unsupervised object segmentation, bridging the gap towards methods that use additional supervision from depth or optical flow.

Our motivation stems from insights into human cognition and an analysis of contemporary self-supervised pre-training techniques. We emphasise a promising avenue for representation that exhibits a deeper understanding of the 3D structure of the world. A potential area for future exploration lies in the development of more robust pre-training methods that encapsulate both semantic and geometric information within the visual scene, as exemplified by DINOv2 [207].

Conversely, when efficiently leveraging a pre-trained visual model, we bypass the challenge of end-to-end learning of hierarchical representations. Without a static, pre-trained vision model, it is currently infeasible to concurrently train object and visual representations using the same architecture and data. Drawing parallels to a prior era when end-to-end training of deep models became both effective and efficient, we find layer-wise pre-training falling out of favour within the research community. A similar shift may apply to object-centric and visual learning, whereby developing a methodology for their joint end-to-end training would yield a significantly more capable and potent system.

Chapter 8

Conclusions

In this thesis, we provided several contributions on the topic of representation learning within the domain of computer vision, with the aim of contributing to the advancement and development of methods and systems that can learn to perceive and reason better. We addressed **Research Question 1**, on learning generally useful visual representation of the world by covering two complementary directions to advance the field of representation learning, as succinctly captured in the **Scaling hypothesis** and the **Structured Representation** hypothesis, introduced in Section 1.

The **Scaling hypothesis** emphasises the importance of scale in computation power and data, especially for learning and searching methods. In the case of deep neural networks, this translates to both in the number of parameters of a neural networks and the data on which they were trained. In Chapter 3, we provided an extensive review of a general framework for Contrastive Representation Learning as a promising direction for learning generally useful representations on a wide variety of tasks, unconstrained by the limitations of human-labelled datasets. This method directly enables scaling up neural networks to utilise unprecedented amounts of compute and data.

The **Structured Representation** hypothesis emphasises capturing the underlying structure of the data generation process, as a fundamental step to propel deep neural networks from the domain of perception to reasoning tasks. Object-centric Representation Learning is a promising approach in that regard, aiming to simultaneously discover and represent objects in a complex scene in an independent manner. We introduced and carried out various experiments surrounding this approach for learning structured representations from Chapter 4 to Chapter 7.

We now restate and highlight our contributions in this thesis as well as provide an opinion on the current state of research and future directions.

8.1 Contrastive Representation Learning: A Framework and Review

In the realm of representation learning, numerous self-supervised methods have emerged, yet their downstream performances often fall short when compared to supervised learning settings. However, within this landscape, a subset of methods, collectively known as Contrastive Learning, have demonstrated the ability to outperform traditional supervised learning methods.

Although there has been a recent surge of interest in the topic, contrastive learning and contrastively learning representations is not a new idea, with work dating back nearly 30 years to the early 1990s. This is partly because much of the machine learning field is now taken up by problems of data architecture and systems engineering and scalability. This usually involves building systems which are bigger and operating under the maxim that bigger is better. Contrastive learning is more like data engineering and it allows the properties of data to emerge naturally based on data similarity rather than trying to fit data processing into some large and complex system architecture.

To systematically study the fast growing topic of contrastive learning, we conducted a thorough review, analysing over 100 methods across various data modalities. Our contribution is a well-defined framework that categorises these methods based on key components: the data similarity distribution, the encoder, the transform head, and the contrastive loss. This framework serves as a valuable tool, facilitating a deeper understanding of the fundamental principles, historical development, and rationale behind contrastive learning, while also providing a means of comprehending the contributions of new methods.

Because contrastive learning has been used in multiple applications and input domains including image, video, text, audio and others, we have had to draw together input from NLP, computer vision, audio processing and more in order to present a comprehensive survey of the field, with inputs also drawn from across these disparate application areas. Our exploration covered the entire contrastive learning framework, spanning diverse data domains, and culminated in the development of a taxonomy of approaches for each individual component. While our focus has been on contrastive learning, these taxonomies extend beyond this, offering general principles applicable to the implementation of inductive biases in all self-supervised learning systems.

While the chapter will provide a useful resource for those who have little background in the topic of contrastive learning and who want to learn more, it will also be of value to those already familiar with the topic since contributions to the development of the area are drawn from such a range of sources.

Contrastive learning and contrastive representations of data represent an interesting and different approach to modeling data which is suited to some kinds of datasets, and for applications where labelled training data may not be available or in sufficient amounts to support typical deep learning approaches.

Whilst successful contrastive representation learning typically involves using relatively more computational resources (and thus power), the models produced by this process often enable rich general-purpose representations that show greater performance on a variety of downstream tasks than their end-to-end counterparts. Ultimately, this may result in less computational resources being consumed when using pre-trained contrastive representation models for as basis for new tasks.

Contrastive learning is not a panacea for all kinds of problems in data modeling and data classification, prediction and clustering, but for a reasonable subset of application types, on certain types of datasets it is a suitable approach to improve performance on downstream tasks. Nor is it an approach with all of its problems and issues solved, and in chapter 3 we highlighted areas for future research, some of which are fundamental issues.

One of the promising aspects of contrastive learning is its synergy with other approaches, such as incremental or lifelong learning, which are essential in the pursuit of creating generally useful AI agents capable of operating in real-world environments. Contrastive learning can serve as a robust pre-trained foundation for incremental learning methods, enabling models to adapt to new tasks without forgetting previously learned information [169]. Additionally, it can be directly applied to the incremental discovery of objects, facilitating the continual learning process as new objects or patterns emerge in the data [298]. By integrating contrastive learning with incremental learning strategies, we move closer to developing AI systems that are both flexible and resilient, capable of ongoing adaptation and improvement in dynamic settings.

For practitioners who want to apply contrastive methods for pre-training representations on different datasets, we suggest to be mindful about:

- Any inherent characteristics and biases in the data set, e.g. do the images contain only one or multiple objects, are the objects in the center, etc.
- The desired properties of the representation for downstream tasks, e.g. occlusion-invariance, colour-invariance, temporal-covariance, etc.
- The ways positive and negative pairs are constructed, such that they provide good learning signals and convey the desired properties.

Using the CRL framework, this chapter addressed **Research Question 2** by investigating the general principles and inductive biases for learning such representations.

8.2 Review and Appraisal of Object-centric Representations

In this section,we take the perspectives outlined in the previous chapters on the future direction of object-centric representational learning, and review these in the context of the overall contributions of the thesis.

In Chapter 4, we introduced the topic of Object-centric representation learning, its motivation, goals, and development over recent years. Among these methods, learning object representations via the slot attention mechanism, as outlined in SAVi, is the most efficient and achieves state-of-the-art results. We explored and proposed various methods at different stages of the pipeline for OCRL, from architectural biases in the representation format (discrete) to practical challenges in efficiently learning object-centric models (slot decoder), and leveraging geometric signals from pre-trained models. Together with advances in the broader field of deep learning, we have witnessed improvements in methods and challenges, starting from simple datasets and progressing to more challenging ones.

Object-centric Representation Learning: We then turned to the challenges of learning structured representation as stated in **Research Question 3**. We first presented in Chapter 4 an overview of the problem setting, the motivation as well as the foundational work on the topic of Object-centric Representation Learning. We also presented the learning framework for learning slot-based object-centric representations of videos, which served as the backbone for our subsequent experiments.

Discrete Object-centric Representation: In Chapter 5, we proposed a novel method to learn a discrete object-centric representation space, based on the framework of Vector Quantisation. This was motivated by the inherent discrete nature of objects that we are trying to capture and the inductive bias of the quantisation technique.

We showcased the feasibility of our approach by integrating and comparing it with a state-of-the-art object-centric representation learning method designed for video datasets. Despite the more restrictive nature of the latent space, our method performs on par with the continuous-representation baseline on the object-discovery task.

Attentional Slot Decoder: In Chapter 6, we proposed a simple Attentional Slot Decoder for object-centric representation learning methods in the framework of autoencoding. Current methods based on reconstruction-based object-centric learning, which comprises the majority of work, all require decoding each object's

slot representations independently, and merging their outputs at the pixel level. This is computationally expensive and does not allow for rich interaction of objects at the representation level.

Our simple approach combines the strengths of both slot-based decoding and the more general set-based decoding. The core idea of our method is the utilisation of a cross attention module between the positional decoder query and the object representations. This rich interaction in the latent space allows for the exchange of semantic object information.

All else being equal, our proposed decoder learns faster than the baseline, and achieves comparable performance while requiring substantially less memory and compute time.

This chapter directly addressed the efficiency aspect of learning structured representation as posed in **Research Question 4**.

Object Discovery with Geometric Representation: In this chapter, our focus shifted from architecture designs in learning object-centric representation, as discussed in the previous chapters, to the learning signals used for object-centric representation. By harnessing a pre-trained vision model enriched with geometric information in its latent space, we established a comprehensive pipeline for unsupervised object segmentation, bridging the gap between methods that use additional supervision from depth or optical flow. Together, our approach is modular and remains completely free of the need for human supervision.

Now, the field is at a point where it has generally capable object-centric models, but their application on realistic datasets is still limited. Scaling these methods to work with more complex, realistic, and diverse datasets is the most important challenge for the field at this time.

8.2.1 Review of Recent Progress

In this section, we explore some different aspects of object centric representation learning, from alternative representation formats and problem settings, and discuss the future research direction.

Representation format: In the same vein as our exploration of Discrete Object-centric Representations in Chapter 5, investigating alternative formats for representing objects remains an active research problem.

Block Slot Attention [248] aims to enhance the disentanglement aspect of object slots by decomposing each slot into several blocks from a common concept memory bottleneck.

Contrary to popular slot representations approaches that are discrete in nature, complex-valued autoencoders [173] propose learning continuous and distributed representations. Generalising from a set of real-valued vectors, representations for all objects are encoded in complex-valued vectors, where the magnitude indicates the presence of features, and the relative phase is used to group features into objects. This idea is further extended to more dimensions by Rotating Features [175] that shows strong performance on the object discovery task with real images.

Learning efficiency Tackling the object decoder component, as explored in our presentation of the Attentional Slot Decoder in Chapter 6, Slot Diffusion [290] incorporates the powerful Latent Diffusion Model [227] for image generation into object-centric learning. They demonstrate strong performance results on both object discovery and generation, leveraging the powerful visual modelling capability of the pretrained diffusion model.

Object discovery in 3D: In Chapter 7, we emphasised the importance of 3D geometry in enabling object discovery and learning in more complex settings. Besides image and video-based methods, explicitly learning object-centric representations in a 3D setting is another interesting direction of work.

Leveraging recent advances in neural 3D scene representation with Neural Radiance Field (NERF) [188], several recent works [303, 258, 200] propose learning to decompose 3D scene representations with object-centric representations.

In addition to image and video-based learning, OSRT [234] learns an object-factored 3D scene representation from multiple views of a static scene. DORSal [127] adapts a video diffusion model with the object slots from OSRT to achieve scalable object-level scene rendering.

Reconstruction-free training: In addition to generative approaches based on reconstruction or novel view synthesis objectives, end-to-end contrastive training of object representations is a promising alternative [149]. Löwe et al. [174] extends the per-slot objective to a global set-based contrastive loss.

ODIN [112] is another reconstruction-free approach that learns to simultaneously discover and represent objects via a contrastive loss based on multi-crop and image augmentation. Conversely, Wang, Shou, and Zhang [282] imposes a cyclical consistency between object-centric representations and visual features to learn and discover objects without a contrastive or reconstruction objective.

Understanding and optimisation: In addition to advancing the state of the art, many works aim to understand the learning and optimisation dynamics of object-

centric learning. Chang, Levine, and Griffiths [36] proposes to examine the problem of learning hierarchical representations of objects on top of visual features from the viewpoint of nested optimisation. This perspective leads to various approaches that aid in learning and optimising the iterative refinement of object-centric representations [132, 35].

Prabhudesai et al. [218] proposes to adapt object representation per scene at test time via a self-supervised objective. EGO [311] provides a simple framework for object-centric learning through energy-based models. On the other hand, Brady et al. [25] studies the theoretical conditions in which compositional learning of object representations is possible.

8.2.2 Benefits and Applications of Object-centric Representations

The field of object-centric representation learning has so far been focusing mostly on advancing the state of the art in methods for unsupervised segmentation and representations of objects. However, these are just the means to an end. The ultimate goal is to learn a representation space at a level of abstraction that facilitates more efficiency in learning, more robustness to noise, and one which can generalise in a more systematic manner such that it is useful and can transfer to a range of downstream tasks.

There have been some studies [65, 256] on the generalisation and robustness properties of object representations in Out-of-Distribution settings. Slot representations that can segment objects more accurately also perform better for downstream tasks. These representations are also more robust to certain settings of distribution shifts in the underlying data and enable downstream reinforcement learning agents to achieve their goals when compared to an agent that uses a conventional scene-level representation.

Zhang, Gupta, and Zisserman [309] studies the transferability of object-centric representations and shows that it is more beneficial in various settings from novel objects, few-shot learning, linear probing as well as standard classification settings.

Aloe [64] proposes a method that applies "attention over learned object embedding" with self-supervision for downstream visual question answering tasks. They show that this matches or exceeds the performance of previous state-of-the-art hybrid or fully neural networks with less training on a variety of benchmarks covering object-permanence [83], explanatory, predictive and counterfactual reasoning [301] as well as causal inference [308].

Mambelli et al. [178] demonstrates reinforcement learning methods for object-manipulation that use object-centric representations that can generalise to zero-shot

settings when the number of objects in a scene change.

The SLotFormer method [291] improves performance on planing and visual question-answering tasks when using an object-centric dynamic model over time.k More recently, object-centric representations from [234] successfully improve control and planning even in large-scale foundation models for robot PaLM-E [69].

These studies have validated the synergy between the segregational, representational and compositional aspects of object-centric representations, whilst demonstrating their benefit on various downstream applications. This complements and provides evidence for the usefulness of our work in previous chapters to advance the state-of-the-art in learning object-centric representations.

8.2.3 Challenges and Future Research Directions

Despite the significant progress made in recent years, as outlined in Section 8.2.1, the field of object-centric learning has a wide range of open questions and challenges on its path to becoming more successful and widely applicable in the broader realm of computer vision and deep learning. In this section, we explore some of the potential avenues for future research, as well as taking a step back to revisit some of the underlying assumptions in the field.

Exploring Alternative Directions While slot-based representation has been the prevalent choice for learning object representations, primarily due to its natural extension from visual representations, it is essential to consider whether there might be more suitable methods for simultaneously discovering and learning objects. What are the alternative formats beyond slot-based representations for object-centric learning? In addition to the complex-valued [173] and rotating features [175] as mentioned earlier, Tensor Product Representation [250] and temporal codes based on spiking neurons [246] have been proposed but remain underexplored in the current literature of object-centric learning. Exploring these alternatives could offer novel insights into the field.

Bridging the Gap with Specialised Models As we have discussed, recent advancements in object-centric learning have enabled the application of these methods in real-world, visually complex scenes with minimal supervision. However, there still exists a performance gap when compared to neural networks specifically trained for explicit object detection tasks. Closing this gap and moving beyond pure detection to simultaneously discovering and representing objects is a crucial research direction, where developments from the field of unsupervised object localisation [280, 245, 281] could provide valuable lessons for object-centric learning approaches. In-

vestigating how these techniques can enhance the capabilities and be incorporated into object-centric models is an important avenue for future research.

The Role and Effect of Human Labels While unsupervised object discovery through self-supervision has been a fundamental aspect of object-centric learning, there is potential in directly incorporating models trained for human-supervised object detection. The field of computer vision has long grappled with the debate between data-driven learning and inductive bias in architectures. Should self-supervised architectures only rely on aspects such as visual similarity, temporal correlation, spatial and temporal locality, or shared functionality for object discovery?

Supervised methods for object detection and segmentation, like Segment Anything Model (SAM) [152], have demonstrated the benefits of a large-scale, data-driven approach to these perception tasks. The question arises: can we integrate human labels into the object-centric learning loop, and if so, how do we strike a balance between a data-driven approach and learning the inductive biases that define an object? If such supervised models are developed, how will the incorporation of human labels at scale impact the performance and characteristics of object-centric learning methods? This exploration into the role and influence of human-labelled data in object-centric learning is a significant avenue for future research.

Controllable and Steerable Representations While supervised labels and unsupervised inductive biases contribute to segregation and representation in object-centric learning, this addresses only part of the puzzle. The utility of composing object-centric representations to tackle more challenging tasks ultimately depends on the current context and the nature of the task itself.

Approaches like SAVi [150] have taken the initial step of allowing slot initialisation to be conditioned on additional information such as object bounding boxes or masks. Extending this approach to enable more fine-grained control of object segregation and composition, such as incorporating natural language, is another crucial direction. While these methods currently operate by conditioning from the bottom up, incorporating top-down feedback, as in the case of Reasoning Modulated Representation [275], provides an equally important direction. These approaches could allow leveraging interaction data from robots to enable embodied object-centric representations, combined with large language models for both discovery and reasoning capabilities.

Reconciling End-to-End Learning and Modular Structure Another fundamental aspect of object-centric learning is the emphasis on end-to-end learning of

hierarchical representations. An end-to-end system could facilitate both bottom-up and top-down feedback across all stages, from segregation and representation to composition. However, recent advances have relied on powerful pre-trained models to scale up object-centric methods, as demonstrated in [240] and our work in Chapter 7, thus effectively bypassing the challenge of end-to-end learning. How to reconcile and combine the benefits of modular and pre-trained systems versus end-to-end learning remains another open area for exploration?

Specific Architectures for Object-centric Learning? In examining the broader landscape surrounding object-centric representation learning, it is crucial to recall its foundational principle rooted in the hierarchical representation thesis [40]. This framework is motivated by a focus on a level of abstraction similar to what humans effortlessly perceive as "objects", with the ultimate goal of developing a more generally capable intelligent system, agent, or model that can perceive, reason, and plan.

Recent progress in large-scale pre-training, especially with large language models and foundational multimodal models combining images and text, has demonstrated elementary reasoning capabilities, albeit with some brittleness [123]. Notably, these capabilities emerged without the need for specialised architecture or domain-specific components, but relied on large-scale data and model sizes. Another example is recent work showing that object-like representations can emerge simply by scaling up and adding extra "register" tokens during the training of standard Vision Transformers [54]. Building on lessons learned from Slot Attention, [288] also demonstrated that object-centric representations can emerge with minimal adaptation to the widely popular standard transformer architecture used in other deep learning domains.

While developing more specialised architectures for object-centric representation learning, with additional architectural biases, has shown promising results in small-scale and limited settings, it must be approached carefully to avoid hindering the scalability of the entire deep learning pipeline. This raises the the question of whether developing specialised architectures for object-centric learning is worthwhile in the long term. The answer remains uncertain, and the methods developed for object-centric learning might not ultimately be universally adopted. However, an undeniable aspect is that the study and development of methods that learn more abstract, hierarchical representations contribute significantly to advancing the entire field of representation learning and deep learning in general.

8.3 Constraints and Limitations

Throughout this research, several constraints and limitations characteristic of deep learning at the present time have been encountered. These challenges have influenced the direction, scope, and outcomes of this work. To contextualize the contributions of this thesis within the broader landscape, the following major constraints are outlined:

Computational Resources Deep learning research demands substantial computational power, often requiring advanced hardware such as GPUs or TPUs. In particular, scaling self-supervised methods like contrastive learning typically involves training models using hundreds of GPUs over extended periods. However, the compute resources available during this research, generally limited to a single GPU, posed significant challenges. This constraint impacted the speed of experimentation, the feasibility of exploring more complex models, and the overall research direction. The necessity to optimize within these limitations often dictated the scale and ambition of the methodologies explored.

Data Scarcity and Quality As models increase in complexity and scale, the availability of high-quality, annotated datasets becomes increasingly critical. This research faced challenges in accessing such datasets due to several factors, including privacy concerns, proprietary restrictions, and the considerable cost and time required for data collection and annotation. The scarcity of large, diverse datasets limited the ability to train and validate models comprehensively, often necessitating the use of smaller, potentially biased datasets that may affect the generalizability of the results. This is also a main factor in the usage of synthetic data in our experiments.

Collaboration and Organizational Constraints Deep learning is a rapidly evolving field, with significant interest and investment from industry. Large-scale projects often involve extensive collaboration, with dozens of authors and numerous research engineers in the background contributing to the development and fine-tuning of models. In contrast, academic settings, particularly at the PhD level in our case, offer more limited opportunities for collaboration and organizational support. The lack of access to large, multidisciplinary teams and the absence of dedicated research engineers made it challenging to replicate the level of innovation and complexity seen in industry-driven research.

Challenges of a Rapidly Evolving Field The fast-paced nature of deep learning research imposes considerable pressure to stay abreast of the latest advance-

ments. The rapid evolution of state-of-the-art methods, coupled with the high standards required for publication, presents a significant challenge. Keeping up with relevant work is demanding, as is ensuring that research contributions remain timely and impactful. Additionally, reproducing existing research can be difficult due to differences in hardware, software versions, and random model initialization. Such variability can lead to inconsistent results, complicating efforts to validate and extend upon prior work.

These constraints have been integral in shaping the trajectory of this research, influencing both the methodological choices and the overall impact of the work. Recognizing and addressing these limitations provides a foundation for future research to advance beyond these challenges.

8.4 Final outlook

Throughout this thesis, we extensively explored the current state and potential future directions in both Contrastive Representation Learning and Object-centric Representation Learning.

One prime example of the relevance of our work to the field can be seen in the rise of multimodal models. The rapid evolution of multimodal models, which integrate diverse data types such as images, text, and audio, represents a significant advancement. These models leverage the strengths of each modality, resulting in more comprehensive and context-aware systems. Contrastive learning method is the foundation method for building a common representation space, bridging between different input modalities such as text and vision. The approach has been shown to work at web-scale data and is widely used for all the nascent Vision Language Models. In addition, our work on treating the representations of other pre-trained models as another data modality aligns with the rising trend of distilling massive general multimodal models from smaller specilised models.

Our approach to the topic of Representation Learning encompasses both content and structure, via the contrastive learning and object-centric learning frameworks, respectively. Historically, these two sub-fields have progressed somewhat independently. Only recently have the trajectories of scaling and structured representation begun to converge.

The synthesis of these two directions — scaling and structured representation — marks a critical juncture. The design of methods capable of learning structured representations at scale emerges as one of the paramount research directions for advancing the field. To do so, we would need to address the friction between the structure and the expressivity of the representation, resolving the challenge of modular neural networks versus end-to-end learning, and last but not least, the balance

between data-driven and inductive biases in representation learning systems.

Bibliography

[1] Triantafyllos Afouras et al. "Self-Supervised Learning of Audio-Visual Objects from Video". Aug. 10, 2020. arXiv: 2008.04237.

[2] Jean-Baptiste Alayrac et al. "Flamingo: A Visual Language Model for Few-Shot Learning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 23716–23736.

[3] *An Update On Our Use of Face Recognition*. Meta. Nov. 2, 2021.

[4] Ankesh Anand et al. "Unsupervised State Representation Learning in Atari". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach et al. 2019, pp. 8766–8779. URL: https://proceedings.neurips.cc/paper/2019/hash/6fb52e71b837628ac16539c1ff911667-Abstract.html.

[5] OpenAI: Marcin Andrychowicz et al. "Learning Dexterous In-Hand Manipulation". In: *The International Journal of Robotics Research* 39.1 (2020), pp. 3–20.

[6] Relja Arandjelovic and Andrew Zisserman. "Look, Listen and Learn". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 609–617. DOI: 10.1109/ICCV.2017.73. URL: https://doi.org/10.1109/ICCV.2017.73.

[7] Relja Arandjelovic and Andrew Zisserman. "Objects That Sound". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 435–451.

[8] Diego Ardila et al. "End-to-End Lung Cancer Screening with Three-Dimensional Deep Learning on Low-Dose Chest Computed Tomography". In: *Nature Medicine* 25.6 (6 June 2019), pp. 954–961. ISSN: 1546-170X. DOI: 10.1038/s41591-019-0447-x.

[9] Mahmoud Assran et al. *Masked Siamese Networks for Label-Efficient Learning*. Apr. 14, 2022. arXiv: 2204.07141 [cs, eess]. preprint.

[10] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer Normalization". 2016. arXiv: 1607.06450.

[11] Philip Bachman, R. Devon Hjelm, and William Buchwalter. "Learning Representations by Maximizing Mutual Information Across Views". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach et al. 2019, pp. 15509–15519. URL: https://proceedings.neurips.cc/paper/2019/hash/ddf354219aac374f1d40b7e760ee5bb7-Abstract.html.

[12] Alexei Baevski, Steffen Schneider, and Michael Auli. "Vq-Wav2vec: Self-Supervised Learning of Discrete Speech Representations". In: International Conference on Learning Representations. Sept. 25, 2019.

[13] Alexei Baevski et al. "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html.

[14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1409.0473.

[15] Randall Balestriero and Yann LeCun. *Contrastive and Non-Contrastive Self-Supervised Learning Recover Global and Local Spectral Embedding Methods*. May 26, 2022. arXiv: 2205.11508 [cs, math, stat]. preprint.

[16] Adrien Bardes, Jean Ponce, and Yann LeCun. "Vicreg: Variance-invariance-covariance Regularization for Self-Supervised Learning". 2021. arXiv: 2105.04906.

[17] Peter W. Battaglia et al. "Interaction Networks for Learning about Objects, Relations and Physics". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. Ed. by Daniel D. Lee et al. 2016, pp. 4502–4510. URL: https://proceedings.neurips.cc/paper/2016/hash/3147da8ab4a0437c15ef51a5cc7f2dc4-Abstract.html.

[18] Peter W. Battaglia et al. "Relational Inductive Biases, Deep Learning, and Graph Networks". Oct. 17, 2018. arXiv: 1806.01261 [cs, stat].

[19] Suzanna Becker and Geoffrey E. Hinton. "Self-Organizing Neural Network That Discovers Surfaces in Random-Dot Stereograms". In: *Nature* 355.6356 (Jan. 1992), pp. 161–163. ISSN: 1476-4687. DOI: 10.1038/355161a0.

[20] Yoshua Bengio. "The Consciousness Prior". Sept. 25, 2017. arXiv: 1709.08568 [cs, stat].

[21] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation Learning: A Review and New Perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828. DOI: 10.1109/tpami.2013.50. arXiv: 1206.5538.

[22] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation". Aug. 15, 2013. arXiv: 1308.3432 [cs].

[23] Rishi Bommasani et al. "On the Opportunities and Risks of Foundation Models". Aug. 18, 2021. arXiv: 2108.07258 [cs].

[24] Antoine Bordes et al. "Translating Embeddings for Modeling Multi-relational Data". In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. Ed. by Christopher J. C. Burges et al. 2013, pp. 2787–2795. URL: https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html.

[25] Jack Brady et al. *Provably Learning Object-Centric Representations*. May 23, 2023. arXiv: 2305.14229 [cs]. preprint.

[26] Jane Bromley et al. "Signature Verification Using a "Siamese" Time Delay Neural Network". In: (Feb. 1993), p. 8.

[27] Rodney A. Brooks. "Intelligence without Representation". In: *Artificial intelligence* 47.1-3 (1991), pp. 139–159. DOI: 10.1016/0004-3702(91)90053-m.

[28] Christopher P. Burgess et al. "MONet: Unsupervised Scene Decomposition and Representation". Jan. 22, 2019. arXiv: 1901.11390 [cs, stat].

[29] Nick Cammarata et al. "Thread: Circuits". In: *Distill* 5.3 (2020), e24.

[30] Shuhao Cao, Peng Xu, and David A. Clifton. *How to Understand Masked Autoencoders*. Feb. 9, 2022. arXiv: 2202.03670 [cs]. preprint.

[31] Nicolas Carion et al. "End-to-End Object Detection with Transformers". In: *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[32] Mathilde Caron et al. “Deep Clustering for Unsupervised Learning of Visual Features”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 132–149. arXiv: 1807.05520.

[33] Mathilde Caron et al. “Emerging Properties in Self-Supervised Vision Transformers”. Apr. 29, 2021. arXiv: 2104.14294 [cs].

[34] Mathilde Caron et al. “Unsupervised Learning of Visual Features by Contrasting Cluster Assignments”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/70feb62b69f16e0238f741fab228fec2-Abstract.html.

[35] Michael Chang, Thomas L. Griffiths, and Sergey Levine. *Object Representations as Fixed Points: Training Iterative Refinement Algorithms with Implicit Differentiation*. July 2, 2022. arXiv: 2207.00787 [cs]. preprint.

[36] Michael Chang, Sergey Levine, and Thomas L Griffiths. “OBJECT-CENTRIC LEARNING AS NESTED OPTIMIZATION”. In: (2022), p. 7.

[37] Gal Chechik et al. “Large Scale Online Learning of Image Similarity through Ranking”. In: *Pattern Recognition and Image Analysis*. Ed. by Helder Araujo et al. Berlin, Heidelberg: Springer Berlin Heidelberg, June 12, 2009, pp. 11–14. DOI: 10.1007/978-3-642-02172-5_2.

[38] Liang-Chieh Chen et al. “Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected Crfs”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), pp. 834–848.

[39] Lili Chen et al. “Decision Transformer: Reinforcement Learning via Sequence Modeling”. In: *Advances in neural information processing systems* 34 (2021), pp. 15084–15097.

[40] Minshuo Chen et al. “Towards Understanding Hierarchical Learning: Benefits of Neural Representations”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 22134–22145.

[41] Ting Chen et al. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1597–1607. URL: http://proceedings.mlr.press/v119/chen20j.html.

[42] Ting Chen et al. “A Unified Sequence Interface for Vision Tasks”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 31333–31346.

[43] Ting Chen et al. “Big Self-Supervised Models are Strong Semi-Supervised Learners”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/fcbc95ccdd551da181207c0c1400c65Abstract.html.

[44] Xinlei Chen and Kaiming He. “Exploring Simple Siamese Representation Learning”. In: *Computer Vision and Pattern Recognition* (2020). DOI: 10.1109/CVPR46437.2021.01549.

[45] Zewen Chi et al. “InfoXLM: An Information-Theoretic Framework for Cross-Lingual Language Model Pre-Training”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, 2021, pp. 3576–3588. DOI: 10.18653/v1/2021.naacl-main.280. URL: https://aclanthology.org/2021.naacl-main.280.

[46] Kyunghyun Cho et al. “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179. URL: https://aclanthology.org/D14-1179.

[47] Sumit Chopra, Raia Hadsell, and Yann LeCun. “Learning a Similarity Metric Discriminatively, with Application to Face Verification”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 1. IEEE, 2005, pp. 539–546. DOI: 10.1109/cvpr.2005.202.

[48] Dan Claudiu Ciresan et al. “Flexible, High Performance Convolutional Neural Networks for Image Classification”. In: *Twenty-Second International Joint Conference on Artificial Intelligence*. Citeseer, 2011.

[49] Ronan Collobert and Jason Weston. “A unified architecture for natural language processing: deep neural networks with multitask learning”. In: *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*. Ed. by William W. Cohen, Andrew McCallum, and Sam T. Roweis. Vol. 307. ACM International Conference Proceeding Series. ACM, 2008, pp. 160–167. DOI: 10.1145/1390156.1390177. URL: https://doi.org/10.1145/1390156.1390177.

[50] Nelson Cowan. "Working Memory Underpins Cognitive Development, Learning, and Education". In: *Educational psychology review* 26 (2014), pp. 197–223.

[51] Eric Crawford and Joelle Pineau. "Spatially Invariant Unsupervised Object Detection with Convolutional Neural Networks". In: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, pp. 3412–3420. DOI: 10.1609/aaai.v33i01.33013412. URL: https://doi.org/10.1609/aaai.v33i01.33013412.

[52] Marco Cuturi. "Sinkhorn Distances: Lightspeed Computation of Optimal Transport". In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. Ed. by Christopher J. C. Burges et al. 2013, pp. 2292–2300. URL: https://proceedings.neurips.cc/paper/2013/hash/af21d0c97db2e27e13572cbf59eb343Abstract.html.

[53] Tri Dao et al. "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness". In: *Advances in Neural Information Processing Systems* 35 (Dec. 6, 2022), pp. 16344–16359.

[54] Timothée Darcet et al. *Vision Transformers Need Registers*. Sept. 28, 2023. arXiv: 2309.16588 [cs]. preprint.

[55] Guy Davidson and Brenden M. Lake. *Investigating Simple Object Representations in Model-Free Deep Reinforcement Learning*. May 28, 2020. DOI: 10.48550/arXiv.2002.06703. arXiv: 2002.06703 [cs, stat]. preprint.

[56] Jeffrey De Fauw et al. "Clinically Applicable Deep Learning for Diagnosis and Referral in Retinal Disease". In: *Nature Medicine* 24.9 (9 Sept. 2018), pp. 1342–1350. ISSN: 1546-170X. DOI: 10.1038/s41591-018-0107-6.

[57] Mostafa Dehghani et al. "Scaling Vision Transformers to 22 Billion Parameters". In: *International Conference on Machine Learning*. PMLR, 2023, pp. 7480–7512.

[58] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848. URL: https://doi.org/10.1109/CVPR.2009.5206848.

[59] Guilherme N. DeSouza and Avinash C. Kak. “Vision for Mobile Robot Navigation: A Survey”. In: *IEEE transactions on pattern analysis and machine intelligence* 24.2 (2002), pp. 237–267.

[60] Tim Dettmers et al. “LLM.Int8 (): 8-Bit Matrix Multiplication for Transformers at Scale”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 30318–30332.

[61] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[62] Sindhu Devunooru et al. “Deep Learning Neural Networks for Medical Image Segmentation of Brain Tumours for Diagnosis: A Recent Review and Taxonomy”. In: *Journal of Ambient Intelligence and Humanized Computing* 12 (2021), pp. 455–483.

[63] Prafulla Dhariwal et al. “Jukebox: A Generative Model for Music”. In: *PREPRINT* (2020).

[64] David Ding et al. “Attention over Learned Object Embeddings Enables Complex Visual Reasoning”. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021.

[65] Andrea Dittadi et al. *Generalization and Robustness Implications in Object-Centric Learning*. May 22, 2022. arXiv: 2107.00637 [cs, stat]. preprint.

[66] Monroe D. Donsker and SR Srinivasa Varadhan. “Asymptotic Evaluation of Certain Markov Process Expectations for Large Time, I”. In: *Communications on Pure and Applied Mathematics* 28.1 (1975), pp. 1–47. DOI: 10.1002/cpa.3160280102.

[67] Alexey Dosovitskiy et al. “An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[68] Alexey Dosovitskiy et al. “Discriminative Unsupervised Feature Learning with Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. Ed. by Zoubin Ghahramani et al. 2014, pp. 766–774. URL: https://proceedings.

neurips.cc/paper/2014/hash/07563a3fe3bbe7e3ba84431ad9d055af-Abstract.html.

[69] Danny Driess et al. *PaLM-E: An Embodied Multimodal Language Model*. Mar. 6, 2023. arXiv: 2303.03378 [cs]. preprint.

[70] Debidatta Dwibedi et al. "Learning Actionable Representations from Visual Observations". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Aug. 2, 2018, pp. 1577–1584. arXiv: 1808.00928.

[71] Gamaleldin F. Elsayed et al. *SAVi++: Towards End-to-End Object-Centric Learning from Real-World Videos*. June 15, 2022. arXiv: 2206.07764 [cs]. preprint.

[72] Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. *GENESIS-V2: Inferring Unordered Object Representations without Iterative Refinement*. Jan. 25, 2022. DOI: 10.48550/arXiv.2104.09958. arXiv: 2104.09958 [cs, stat]. preprint.

[73] Martin Engelcke et al. "GENESIS: Generative Scene Inference and Sampling with Object-Centric Latent Representations". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=BkxfaTVFwH.

[74] Aleksandr Ermolov et al. "Whitening for Self-Supervised Representation Learning". In: *International Conference on Machine Learning* (2020).

[75] S. M. Ali Eslami et al. "Attend, Infer, Repeat: Fast Scene Understanding with Generative Models". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. Ed. by Daniel D. Lee et al. 2016, pp. 3225–3233. URL: https://proceedings.neurips.cc/paper/2016/hash/52947e0ade57a09e4a1386d08f17b656-Abstract.html.

[76] Patrick Esser, Robin Rombach, and Bjorn Ommer. "Taming Transformers for High-Resolution Image Synthesis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12873–12883.

[77] Hongchao Fang et al. "CERT: Contrastive Self-supervised Learning for Language Understanding". June 18, 2020. arXiv: 2005.12766.

[78] Xavier Favory et al. "COALA: Co-Aligned Autoencoders for Learning Semantically Enriched Audio Representations". July 8, 2020. arXiv: 2006.08386.

[79] Kunihiko Fukushima. “Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position”. In: *Biological Cybernetics* 36.4 (Apr. 1, 1980), pp. 193–202. ISSN: 1432-0770. DOI: 10.1007/BF00344251.

[80] Marta Garnelo and Murray Shanahan. “Reconciling Deep Learning with Symbolic Artificial Intelligence: Representing Objects and Relations”. In: *Current Opinion in Behavioral Sciences*. Artificial Intelligence 29 (Oct. 1, 2019), pp. 17–23. ISSN: 2352-1546. DOI: 10.1016/j.cobeha.2018.12.010.

[81] Robert Geirhos et al. “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness”. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=Bygh9j09KX.

[82] Justin Gilmer et al. “Neural Message Passing for Quantum Chemistry”. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML’17. Sydney, NSW, Australia: JMLR.org, Aug. 6, 2017, pp. 1263–1272.

[83] Rohit Girdhar and Deva Ramanan. “CATER: A Diagnostic Dataset for Compositional Actions and TEmporal Reasoning”. In: Apr. 4, 2020. arXiv: 1910.04744 [cs].

[84] Rohit Girdhar et al. “OmniMAE: Single Model Masked Pretraining on Images and Videos”. In: (), p. 18.

[85] Yuan Gong, Yu-An Chung, and James Glass. “AST: Audio Spectrogram Transformer”. In: *Proc. Interspeech 2021*. 2021, pp. 571–575. DOI: 10.21437/Interspeech.2021-698.

[86] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[87] Ian J. Goodfellow et al. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. Ed. by Zoubin Ghahramani et al. 2014, pp. 2672–2680. URL: https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

[88] Daniel Gordon et al. “Watching the World Go By: Representation Learning from Unlabeled Videos”. In: *ArXiv preprint* abs/2003.07990 (2020). URL: https://arxiv.org/abs/2003.07990.

[89] Anirudh Goyal and Yoshua Bengio. "Inductive Biases for Deep Learning of Higher-Level Cognition". Dec. 7, 2020. arXiv: 2011.15091 [cs, stat].

[90] Anirudh Goyal et al. "Coordination Among Neural Modules Through a Shared Global Workspace". Mar. 1, 2021. arXiv: 2103.01197 [cs, stat].

[91] Anirudh Goyal et al. "Recurrent Independent Mechanisms". In: *International Conference on Learning Representations*. 2021.

[92] Priya Goyal et al. "Self-Supervised Pretraining of Visual Features in the Wild". 2021. arXiv: 2103.01988.

[93] Robert M. Gray. "Vector Quantization". In: *IEEE ASSP Magazine* 1 (1984), pp. 4–29.

[94] Klaus Greff, Rupesh Kumar Srivastava, and Jürgen Schmidhuber. *Binding via Reconstruction Clustering*. Jan. 20, 2016. arXiv: 1511.06418 [cs]. preprint.

[95] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. "Neural Expectation Maximization". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 6691–6701. URL: https://proceedings.neurips.cc/paper/2017/hash/d2cd33e9c0236a8c2d8bd3fa91ad3acf-Abstract.html.

[96] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. "On the Binding Problem in Artificial Neural Networks". Dec. 9, 2020. arXiv: 2012.05208 [cs].

[97] Klaus Greff et al. *Kubric: A Scalable Dataset Generator*. Mar. 7, 2022. arXiv: 2203.03570 [cs]. preprint.

[98] Klaus Greff et al. "Multi-Object Representation Learning with Iterative Variational Inference". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 2424–2433. URL: http://proceedings.mlr.press/v97/greff19a.html.

[99] Klaus Greff et al. "Tagger: Deep Unsupervised Perceptual Grouping". In: *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016.

[100] Jean-Bastien Grill et al. “Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html.

[101] Aditya Grover and Jure Leskovec. “node2vec: Scalable Feature Learning for Networks”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. Ed. by Balaji Krishnapuram et al. ACM, 2016, pp. 855–864. DOI: 10.1145/2939672.2939754. URL: https://doi.org/10.1145/2939672.2939754.

[102] Zhaohan Daniel Guo et al. “Neural Predictive Belief Representations”. 2018. arXiv: 1811.06407.

[103] Michael Gutmann and Aapo Hyvärinen. “Noise-Contrastive Estimation: A New Estimation Principle for Unnormalized Statistical Models”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 2010, pp. 297–304.

[104] Michael U. Gutmann and Aapo Hyvärinen. “Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics”. In: *The journal of machine learning research* 13.1 (2012), pp. 307–361.

[105] R. Hadsell, S. Chopra, and Y. LeCun. “Dimensionality Reduction by Learning an Invariant Mapping”. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*. Vol. 2. New York, NY, USA: IEEE, 2006, pp. 1735–1742. DOI: 10.1109/cvpr.2006.100.

[106] Tengda Han, Weidi Xie, and Andrew Zisserman. “Video Representation Learning by Dense Predictive Coding”. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. Sept. 10, 2019.

[107] Kaveh Hassani and Amir Hosein Khas Ahmadi. “Contrastive Multi-View Representation Learning on Graphs”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 4116–4126. URL: http://proceedings.mlr.press/v119/hassani20a.html.

[108] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90. URL: https://doi.org/10.1109/CVPR.2016.90.

[109] Kaiming He et al. “Masked Autoencoders Are Scalable Vision Learners”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16000–16009.

[110] Kaiming He et al. “Momentum Contrast for Unsupervised Visual Representation Learning”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 2020, pp. 9726–9735. DOI: 10.1109/CVPR42600.2020.00975. URL: https://doi.org/10.1109/CVPR42600.2020.00975.

[111] Olivier J. Hénaff. “Data-Efficient Image Recognition with Contrastive Predictive Coding”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 4182–4192. URL: http://proceedings.mlr.press/v119/henaff20a.html.

[112] Olivier J. Hénaff et al. “Object Discovery and Representation Networks”. Mar. 16, 2022. arXiv: 2203.08777 [cs].

[113] Alexander Hermans, Lucas Beyer, and Bastian Leibe. “In Defense of the Triplet Loss for Person Re-Identification”. Nov. 21, 2017. arXiv: 1703.07737.

[114] Geoffrey Hinton. “How to Represent Part-Whole Hierarchies in a Neural Network”. Feb. 24, 2021. arXiv: 2102.12627 [cs].

[115] Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. “Transforming Auto-Encoders”. In: *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*. Springer, 2011, pp. 44–51.

[116] Geoffrey E. Hinton, Sara Sabour, and Nicholas Frosst. “Matrix capsules with EM routing”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=HJWLfGWRb.

[117] R. Devon Hjelm and Philip Bachman. “Representation Learning with Video Deep InfoMax”. July 27, 2020. arXiv: 2007.13278.

[118] R. Devon Hjelm et al. “Learning deep representations by mutual information estimation and maximization”. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=Bklr3j0cKX.

[119] Jonathan Ho et al. *Video Diffusion Models*. June 22, 2022. arXiv: 2204.03458 [cs]. preprint.

[120] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (Nov. 1, 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

[121] Elad Hoffer and Nir Ailon. “Deep Metric Learning Using Triplet Network”. In: *International Workshop on Similarity-Based Pattern Recognition*. Springer, 2015, pp. 84–92. DOI: 10.1007/978-3-319-24261-3_7.

[122] Kurt Hornik. “Approximation Capabilities of Multilayer Feedforward Networks”. In: *Neural Networks* 4.2 (Jan. 1, 1991), pp. 251–257. ISSN: 0893-6080. DOI: 10.1016/0893-6080(91)90009-T.

[123] Jie Huang and K. Chang. “Towards Reasoning in Large Language Models: A Survey”. In: *Annual Meeting of the Association for Computational Linguistics* (2022). DOI: 10.48550/arXiv.2212.10403.

[124] Lawrence Hubert and Phipps Arabie. “Comparing Partitions”. In: *Journal of classification* 2 (1985), pp. 193–218.

[125] Minyoung Huh et al. “Straightening out the Straight-through Estimator: Overcoming Optimization Challenges in Vector Quantized Networks”. In: *International Conference on Machine Learning* (2023). DOI: 10.48550/arXiv.2305.08842.

[126] Gabriel Ilharco et al. “Probing Text Models for Common Ground with Visual Representations”. 2020. arXiv: 2005.00619.

[127] Allan Jabri et al. *DORSal: Diffusion for Object-centric Representations of Scenes et Al*. Oct. 17, 2023. arXiv: 2306.08068 [cs]. preprint.

[128] Paras Jain et al. “Contrastive Code Representation Learning”. July 9, 2020. arXiv: 2007.04973.

[129] Joel Janai et al. “Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art”. In: *Foundations and Trends® in Computer Graphics and Vision* 12.1–3 (July 5, 2020), pp. 1–308. ISSN: 1572-2740, 1572-2759. DOI: 10.1561/0600000079.

[130] Eric Jang, Shixiang Gu, and Ben Poole. "Categorical Reparameterization with Gumbel-Softmax". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=rkE3y85ee.

[131] Siddhant M. Jayakumar et al. "Multiplicative Interactions and Where to Find Them". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=rylnK6VtDH.

[132] Baoxiong Jia, Yu Liu, and Siyuan Huang. *Unsupervised Object-Centric Learning with Bi-Level Optimized Query Slot Attention*. Oct. 17, 2022. arXiv: 2210.08990 [cs]. preprint.

[133] Jindong Jiang et al. "SCALOR: Generative World Models with Scalable Object Representations". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=SJxrKgStDH.

[134] Jianbo Jiao et al. "Self-Supervised Contrastive Video-Speech Representation Learning for Ultrasound". Aug. 14, 2020. arXiv: 2008.06607.

[135] Li Jing et al. "Understanding Dimensional Collapse in Contrastive Self-supervised Learning". In: International Conference on Learning Representations. Oct. 6, 2021.

[136] Justin Johnson et al. "CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 1988–1997. DOI: 10.1109/CVPR.2017.215. URL: https://doi.org/10.1109/CVPR.2017.215.

[137] Rafal Jozefowicz et al. "Exploring the Limits of Language Modeling". Feb. 11, 2016. arXiv: 1602.02410.

[138] Rishabh Kabra et al. "SIMONe: View-Invariant, Temporally-Abstracted Object Representations via Unsupervised Video Decomposition". June 7, 2021. arXiv: 2106.03849 [cs].

[139] Daniel Kahneman. *Thinking, Fast and Slow*. Penguin UK, Nov. 3, 2011. 432 pp. Google Books: oV1tXT3HigoC.

[140] Uday Kamath, John Liu, and James Whitaker. *Deep Learning for NLP and Speech Recognition*. Vol. 84. Springer, 2019.

[141] Jared Kaplan et al. “Scaling Laws for Neural Language Models”. Jan. 22, 2020. arXiv: 2001.08361 [cs, stat].

[142] Laurynas Karazija, Iro Laina, and Christian Rupprecht. “Clevrtex: A Texture-Rich Benchmark for Unsupervised Multi-Object Segmentation”. 2021. arXiv: 2111.10265.

[143] Andrej Karpathy. *Software 2.0*. Medium. Mar. 13, 2021.

[144] Justin Kerr et al. “LERF: Language Embedded Radiance Fields”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2023, pp. 19729–19739.

[145] Prannay Khosla et al. “Supervised Contrastive Learning”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html.

[146] Sameer Khurana, Antoine Laurent, and James Glass. “CSTNet: Contrastive Speech Translation Network for Self-Supervised Speech Representation Learning”. Aug. 5, 2020. arXiv: 2006.02814.

[147] Mingyu Kim et al. “Deep Learning in Medical Imaging”. In: *Neurospine* 16.4 (Dec. 2019), pp. 657–668. ISSN: 2586-6583. DOI: 10.14245/ns.1938396.198. pmid: 31905454.

[148] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014. URL: http://arxiv.org/abs/1312.6114.

[149] Thomas Kipf, Elise van der Pol, and Max Welling. “Contrastive Learning of Structured World Models”. In: International Conference on Learning Representations. Sept. 25, 2019.

[150] Thomas Kipf et al. “Conditional Object-Centric Learning from Video”. Nov. 24, 2021. arXiv: 2111.12594 [cs, stat].

[151] Thomas N. Kipf and Max Welling. “Semi-Supervised Classification with Graph Convolutional Networks”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=SJU4ayYgl.

[152] Alexander Kirillov et al. *Segment Anything*. Apr. 5, 2023. DOI: 10.48550/arXiv.2304.02643. arXiv: 2304.02643 [cs]. preprint.

[153] Alexander Kolesnikov et al. "Big Transfer (Bit): General Visual Representation Learning". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer, 2020, pp. 491–507.

[154] Alexander Kolesnikov et al. *UViM: A Unified Modeling Approach for Vision with Learned Guiding Codes*. May 27, 2022. arXiv: 2205.10337 [cs]. preprint.

[155] Lingpeng Kong et al. "A Mutual Information Maximization Perspective of Language Representation Learning". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=Syx79eBKwr.

[156] Adam R. Kosiorek et al. "Sequential Attend, Infer, Repeat: Generative Modelling of Moving Objects". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by Samy Bengio et al. 2018, pp. 8615–8625. URL: https://proceedings.neurips.cc/paper/2018/hash/7417744a2bac776fabe5a09b21c707a2-Abstract.html.

[157] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. Ed. by Peter L. Bartlett et al. 2012, pp. 1106–1114. URL: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

[158] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. "CURL: Contrastive Unsupervised Representations for Reinforcement Learning". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 5639–5650. URL: http://proceedings.mlr.press/v119/laskin20a.html.

[159] Yann LeCun. "A Path towards Autonomous Machine Intelligence Version 0.9. 2, 2022-06-27". In: *Open Review* 62 (2022).

[160] Yann LeCun and Fu Jie Huang. "Loss Functions for Discriminative Training of Energy-Based Models." In: *AIStats*. Vol. 6. Citeseer, 2005, p. 34.

[161] Yann LeCun et al. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural computation* 1.4 (1989), pp. 541–551.

[162] Yann LeCun et al. “Gradient-Based Learning Applied to Document Recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[163] Junnan Li et al. “Prototypical Contrastive Learning of Unsupervised Representations”. May 11, 2020. arXiv: 2005.04966.

[164] Tsung-Yi Lin et al. “Microsoft Coco: Common Objects in Context”. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[165] Zhixuan Lin et al. “SPACE: Unsupervised Object-Oriented Scene Representation via Spatial Attention and Decomposition”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=rkl03ySYDH.

[166] Seppo Linnainmaa. “Taylor Expansion of the Accumulated Rounding Error”. In: *BIT Numerical Mathematics* 16.2 (1976), pp. 146–160.

[167] Dianbo Liu et al. “Adaptive Discrete Communication Bottlenecks with Dynamic Vector Quantization”. In: *AAAI Conference on Artificial Intelligence* (2022). DOI: 10.1609/aaai.v37i7.26061.

[168] Dianbo Liu et al. “Discrete-Valued Neural Communication”. In: *Advances in Neural Information Processing Systems* 34 (July 10, 2021), pp. 2109–2121.

[169] Mingxuan Liu et al. “Large-scale Pre-trained Models are Surprisingly Strong in Incremental Novel Class Discovery”. In: *arXiv preprint arXiv: 2303.15975* (2023).

[170] Francesco Locatello et al. “Object-Centric Learning with Slot Attention”. June 26, 2020. arXiv: 2006.15055 [cs, stat].

[171] Lajanugen Logeswaran and Honglak Lee. “An efficient framework for learning sentence representations”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=rJvJXZb0W.

[172] Sindy Löwe, Peter O’Connor, and Bastiaan S. Veeling. “Putting An End to End-to-End: Gradient-Isolated Learning of Representations”. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach et al. 2019, pp. 3033–

3045. URL: https://proceedings.neurips.cc/paper/2019/hash/851300ee84c2b80ed40f51ed26d866fc-Abstract.html.

[173] Sindy Löwe et al. "Complex-Valued Autoencoders for Object Discovery". In: *Trans. Mach. Learn. Res.* 2022 (2022).

[174] Sindy Löwe et al. "Learning Object-Centric Video Models by Contrasting Sets". Nov. 20, 2020. arXiv: 2011.10287 [cs].

[175] Sindy Löwe et al. *Rotating Features for Object Discovery*. June 1, 2023. arXiv: 2306.00600 [cs]. preprint.

[176] Zhuang Ma and Michael Collins. "Noise Contrastive Estimation and Negative Sampling for Conditional Models: Consistency and Statistical Efficiency". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 3698–3707. DOI: 10.18653/v1/D18-1405. URL: https://aclanthology.org/D18-1405.

[177] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. "The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=S1jE5L5gl.

[178] Davide Mambelli et al. "Compositional Multi-Object Reinforcement Learning with Linear Relation Networks". Jan. 31, 2022. arXiv: 2201.13388 [cs, stat].

[179] R. Manmatha et al. "Sampling Matters in Deep Embedding Learning". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 2859–2867. DOI: 10.1109/ICCV.2017.309. URL: https://doi.org/10.1109/ICCV.2017.309.

[180] Chengzhi Mao et al. *Discrete Representations Strengthen Vision Transformer Robustness*. Apr. 1, 2022. arXiv: 2111.10493 [cs]. preprint.

[181] Loic Matthey et al. *dSprites: Disentanglement Testing Sprites Dataset*. 2017.

[182] Adrienne Mayor. *Gods and Robots: Myths, Machines, and Ancient Dreams of Technology*. Princeton University Press, 2019.

[183] John McCarthy et al. "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955". In: *AI magazine* 27.4 (2006), pp. 12–12.

[184] Warren S. McCulloch and Walter Pitts. “A Logical Calculus of the Ideas Immanent in Nervous Activity”. In: *The bulletin of mathematical biophysics* 5 (1943), pp. 115–133.

[185] Fabian Mentzer et al. *Finite Scalar Quantization: VQ-VAE Made Simple*. Sept. 27, 2023. arXiv: 2309.15505 [cs]. preprint.

[186] Tomas Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. 2013. arXiv: 1301.3781.

[187] Tomás Mikolov et al. “Distributed Representations of Words and Phrases and their Compositionality”. In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. Ed. by Christopher J. C. Burges et al. 2013, pp. 3111–3119. URL: https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.

[188] Ben Mildenhall et al. *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*. Aug. 3, 2020. arXiv: 2003.08934 [cs]. preprint.

[189] Marvin Minsky and Seymour Papert. “An Introduction to Computational Geometry”. In: *Cambridge tiass., HIT* 479.480 (1969), p. 104.

[190] Ishan Misra and Laurens van der Maaten. “Self-Supervised Learning of Pretext-Invariant Representations”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 2020, pp. 6706–6716. DOI: 10.1109/CVPR42600.2020.00674. URL: https://doi.org/10.1109/CVPR42600.2020.00674.

[191] Andriy Mnih and Koray Kavukcuoglu. “Learning word embeddings efficiently with noise-contrastive estimation”. In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. Ed. by Christopher J. C. Burges et al. 2013, pp. 2265–2273. URL: https://proceedings.neurips.cc/paper/2013/hash/db2b4182156b2f1f817860ac9f409ad7-Abstract.html.

[192] Andriy Mnih and Yee Whye Teh. “A fast and simple algorithm for training neural probabilistic language models”. In: *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012. URL: http://icml.cc/2012/papers/855.pdf.

[193] Sangwoo Mo et al. “Object-Aware Contrastive Learning for Debiased Scene Representation”. In: Advances in Neural Information Processing Systems. Nov. 9, 2021.

[194] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. “Audio-Visual Instance Discrimination with Cross-Modal Agreement”. 2020. arXiv: 2004.12943.

[195] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023.

[196] Kevin P. Murphy. *Probabilistic Machine Learning: An Introduction*. MIT Press, 2022.

[197] Apoorv Nandan and Jithendra Vepa. “Language Agnostic Speech Embeddings for Emotion Classification”. In: *ICML 2020 Workshop SAS* (June 10, 2020).

[198] Arvind Narayanan. *Is GPT-4 Getting Worse over Time?* Mar. 20, 2023.

[199] Allen Newell and Herbert A. Simon. “Computer Science as Empirical Inquiry: Symbols and Search”. In: *ACM Turing Award Lectures*. 2007, p. 1975.

[200] Michael Niemeyer and Andreas Geiger. *GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields*. Apr. 29, 2021. DOI: 10.48550/arXiv.2011.12100. arXiv: 2011.12100 [cs]. preprint.

[201] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. “f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization”. In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. Ed. by Daniel D. Lee et al. 2016, pp. 271–279. URL: https://proceedings.neurips.cc/paper/2016/hash/cedebb6e872f539bef8c3f919874e9d7-Abstract.html.

[202] Keiron O’Shea and Ryan Nash. “An Introduction to Convolutional Neural Networks”. 2015. arXiv: 1511.08458.

[203] Daisuke Okanohara and Jun’ichi Tsujii. “A discriminative language model with pseudo-negative samples”. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 73–80. URL: https://aclanthology.org/P07-1010.

[204] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. “Representation Learning with Contrastive Predictive Coding”. In: *ArXiv preprint* abs/1807.03748 (2018). URL: https://arxiv.org/abs/1807.03748.

[205] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. "Neural Discrete Representation Learning". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 6306–6315. URL: https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

[206] R. OpenAI. "GPT-4 Technical Report". In: *arXiv* (2023), p. 2303.08774.

[207] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. Apr. 14, 2023. arXiv: 2304.07193 [cs]. preprint.

[208] A. Paccanaro and Geoffrey Hinton. "Extracting Distributed Representations of Concepts and Relations from Positive and Negative Propositions". In: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*. Como, Italy: IEEE, 2000, 259–264 vol.2. DOI: 10.1109/ijcnn.2000.857906.

[209] Seymour A. Papert. "The Summer Vision Project". In: (July 1, 1966).

[210] Daniel S. Park et al. "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition". In: Interspeech 2019. ISCA, Sept. 15, 2019, pp. 2613–2617. DOI: 10.21437/interspeech.2019-2680.

[211] Taesung Park et al. "Contrastive Learning for Unpaired Image-to-Image Translation". July 30, 2020. arXiv: 2007.15651.

[212] Nikhil Parthasarathy et al. *Self-Supervised Video Pretraining Yields Human-Aligned Visual Representations*. July 25, 2023. arXiv: 2210.06433 [cs]. preprint.

[213] Mandela Patrick et al. "Multi-Modal Self-Supervision from Generalized Data Transformations". June 5, 2020. arXiv: 2003.04298.

[214] Jonas Pfeiffer et al. *Modular Deep Learning*. Feb. 22, 2023. DOI: 10.48550/arXiv.2302.11529. arXiv: 2302.11529 [cs]. preprint.

[215] Jean Piaget. *The Construction of Reality in the Child*. Routledge, 2013.

[216] Sören Pirk et al. "Online Object Representations with Contrastive Learning". 2019. arXiv: 1906.04312.

[217] Ben Poole et al. "On Variational Bounds of Mutual Information". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 5171–5180. URL: http://proceedings.mlr.press/v97/poole19a.html.

[218] Mihir Prabhudesai et al. *Test-Time Adaptation with Slot-Centric Models*. June 27, 2023. arXiv: 2203.11194 [cs]. preprint.

[219] Senthil Purushwalkam and Abhinav Gupta. "Demystifying Contrastive Self-Supervised Learning: Invariances, Augmentations and Dataset Biases". July 29, 2020. arXiv: 2007.13916.

[220] Jiezhong Qiu et al. "GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training". In: *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*. Ed. by Rajesh Gupta et al. ACM, 2020, pp. 1150–1160. URL: https://dl.acm.org/doi/10.1145/3394486.3403168.

[221] Alec Radford et al. "Improving Language Understanding by Generative Pre-Training". In: (2018).

[222] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: (Feb. 26, 2021).

[223] Aditya Ramesh et al. *Hierarchical Text-Conditional Image Generation with CLIP Latents*. Apr. 12, 2022. DOI: 10.48550/arXiv.2204.06125. arXiv: 2204.06125 [cs]. preprint.

[224] David P. Reichert, Peggy Series, and Amos J. Storkey. "A Hierarchical Generative Model of Recurrent Object-Based Attention in the Visual Cortex". In: *International Conference on Artificial Neural Networks*. Springer, 2011, pp. 18–25.

[225] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 3982–3992. DOI: 10.18653/v1/D19-1410. URL: https://aclanthology.org/D19-1410.

[226] Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. Ed. by Corinna Cortes et al. 2015, pp. 91–99. URL: https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html.

[227] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. Apr. 13, 2022. arXiv: 2112.10752 [cs]. preprint.

[228] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[229] Frank Rosenblatt. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." In: *Psychological review* 65.6 (1958), p. 386.

[230] Aurko Roy et al. "Theory and Experiments on Vector Quantized Autoencoders". 2018. arXiv: 1805.11063.

[231] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning Representations by Back-Propagating Errors". In: *nature* 323.6088 (1986), pp. 533–536.

[232] Stuart J. Russell, Peter Norvig, and Ernest Davis. *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall Series in Artificial Intelligence. Upper Saddle River: Prentice Hall, 2010. 1132 pp.

[233] Sara Sabour et al. "Unsupervised Part Representation by Flow Capsules". In: *International Conference on Machine Learning*. PMLR, 2021, pp. 9213–9223.

[234] Mehdi S. M. Sajjadi et al. *Object Scene Representation Transformer*. June 14, 2022. arXiv: 2206.06922 [cs]. preprint.

[235] Ruslan Salakhutdinov. "Deep learning". In: *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*. Ed. by Sofus A. Macskassy et al. ACM, 2014, p. 1973. DOI: 10.1145/2623330.2630809. URL: https://doi.org/10.1145/2623330.2630809.

[236] Adam Santoro et al. “A simple neural network module for relational reasoning”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 4967–4976. URL: https://proceedings.neurips.cc/paper/2017/hash/e6acf4b0f69f6f6e60e9a815938aa1ff-Abstract.html.

[237] Jürgen Schmidhuber. “Deep Learning in Neural Networks: An Overview”. In: *Neural Networks* 61 (Jan. 1, 2015), pp. 85–117. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2014.09.003.

[238] Steffen Schneider et al. “Wav2vec: Unsupervised Pre-Training for Speech Recognition”. In: *Proc. Interspeech 2019* (2019), pp. 3465–3469. DOI: 10.21437/interspeech.2019-1873. arXiv: 1904.05862.

[239] Florian Schroff, Dmitry Kalenichenko, and James Philbin. “FaceNet: A unified embedding for face recognition and clustering”. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682. URL: https://doi.org/10.1109/CVPR.2015.7298682.

[240] Maximilian Seitzer et al. *Bridging the Gap to Real-World Object-Centric Learning*. Mar. 6, 2023. arXiv: 2209.14860 [cs]. preprint.

[241] Younggyo Seo et al. *Masked World Models for Visual Control*. June 28, 2022. arXiv: 2206.14244 [cs]. preprint.

[242] Pierre Sermanet et al. “Time-Contrastive Networks: Self-supervised Learning from Video”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1134–1141. arXiv: 1704.06888.

[243] Yongliang Shen et al. *HuggingGPT: Solving AI Tasks with ChatGPT and Its Friends in HuggingFace*. Apr. 2, 2023. arXiv: 2303.17580 [cs]. preprint.

[244] Oriane Siméoni et al. “Localizing Objects with Self-Supervised Transformers and No Labels”. 2021. arXiv: 2109.14279.

[245] Oriane Siméoni et al. *Unsupervised Object Localization: Observing the Background to Discover Objects*. Mar. 29, 2023. arXiv: 2212.07834 [cs]. preprint.

[246] Wolf Singer. “Distributed Processing and Temporal Codes in Neuronal Networks”. In: *Cognitive neurodynamics* 3 (2009), pp. 189–196.

[247] Gautam Singh, Fei Deng, and Sungjin Ahn. “Illiterate DALL-E Learns to Compose”. Oct. 27, 2021. arXiv: 2110.11405 [cs].

[248] Gautam Singh, Yeongbin Kim, and Sungjin Ahn. *Neural Block-Slot Representations*. Nov. 2, 2022. arXiv: 2211.01177 [cs]. preprint.

[249] Mannat Singh et al. "The Effectiveness of MAE Pre-Pretraining for Billion-Scale Pretraining". 2023. arXiv: 2303.13496.

[250] Paul Smolensky. "Tensor Product Variable Binding and the Representation of Symbolic Structures in Connectionist Systems". In: *Artificial intelligence* 46.1-2 (1990), pp. 159–216.

[251] Kihyuk Sohn. "Improved Deep Metric Learning with Multi-class N-pair Loss Objective". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. Ed. by Daniel D. Lee et al. 2016, pp. 1849–1857. URL: https://proceedings.neurips.cc/paper/2016/hash/6b180037abbebea991d8b1232f8a8ca9-Abstract.html.

[252] Hyun Oh Song et al. "Deep Metric Learning via Lifted Structured Feature Embedding". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 4004–4012. DOI: 10.1109/CVPR.2016.434. URL: https://doi.org/10.1109/CVPR.2016.434.

[253] Elizabeth S. Spelke and Katherine D. Kinzler. "Core Knowledge". In: *Developmental science* 10.1 (2007), pp. 89–96.

[254] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958.

[255] Aleksandar Stanić, Sjoerd van Steenkiste, and Jürgen Schmidhuber. "Hierarchical Relational Inference". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 11. 2021, pp. 9730–9738.

[256] Aleksandar Stanić et al. *Learning to Generalize with Object-centric Agents in the Open World Survival Game Crafter*. Aug. 5, 2022. arXiv: 2208.03374 [cs]. preprint.

[257] Luke Stark. "Facial Recognition, Emotion and Race in Animated Social Media". In: *First Monday* (Sept. 1, 2018). ISSN: 1396-0466. DOI: 10.5210/fm.v23i9.9406.

[258] Karl Stelzner, Kristian Kersting, and Adam R. Kosiorek. *Decomposing 3D Scenes into Objects via Unsupervised Volume Segmentation*. Apr. 2, 2021. arXiv: 2104.01148 [cs, stat]. preprint.

[259] Adam Stooke et al. “Decoupling Representation Learning from Reinforcement Learning”. Sept. 14, 2020. arXiv: 2009.08319.

[260] Karl Stratos. *Noise Contrastive Estimation*. Mar. 2019.

[261] Chen Sun et al. “Learning Video Representations Using Contrastive Bidirectional Transformer”. 2019. arXiv: 1906.05743.

[262] Fan-Yun Sun et al. “InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization”. In: International Conference on Learning Representations. Sept. 25, 2019. arXiv: 1908.01000.

[263] Pei Sun et al. “Scalability in Perception for Autonomous Driving: Waymo Open Dataset”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 2020, pp. 2443–2451. DOI: 10.1109/CVPR42600.2020.00252. URL: https://doi.org/10.1109/CVPR42600.2020.00252.

[264] Richard S. Sutton. *The Bitter Lesson*. The Bitter Lesson. Mar. 13, 2019.

[265] Yuhta Takida et al. “SQ-VAE: Variational Bayes on Discrete Representation with Self-Annealed Stochastic Quantization”. In: *International Conference on Machine Learning* (2022). DOI: 10.48550/arXiv.2205.07547.

[266] Matthew Tancik et al. “Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains”. June 18, 2020. arXiv: 2006.10739 [cs].

[267] Graham W. Taylor et al. “Learning invariance through imitation”. In: *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*. IEEE Computer Society, 2011, pp. 2729–2736. DOI: 10.1109/CVPR.2011.5995538. URL: https://doi.org/10.1109/CVPR.2011.5995538.

[268] Yonglong Tian, Dilip Krishnan, and Phillip Isola. “Contrastive Multiview Coding”. Oct. 20, 2019. arXiv: 1906.05849.

[269] Yonglong Tian, Dilip Krishnan, and Phillip Isola. “Contrastive Representation Distillation”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=SkgpBJrtvS.

[270] Yonglong Tian et al. “What Makes for Good Views for Contrastive Learning”. May 20, 2020. arXiv: 2005.10243.

[271] Michael Tschannen et al. “On Mutual Information Maximization for Representation Learning”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=rkxoh24FPH.

[272] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[273] Petar Veličković and Charles Blundell. “Neural Algorithmic Reasoning”. In: *Patterns* 2.7 (July 2021), p. 100273. ISSN: 26663899. DOI: 10.1016/j.patter.2021.100273. arXiv: 2105.02761 [cs, math, stat].

[274] Petar Veličković et al. “Deep Graph Infomax”. In: International Conference on Learning Representations. Sept. 27, 2018.

[275] Petar Veličković et al. *Reasoning-Modulated Representations*. July 19, 2021. arXiv: 2107.08881 [cs, stat]. preprint.

[276] Jiang Wang et al. “Learning Fine-Grained Image Similarity with Deep Ranking”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE Computer Society, 2014, pp. 1386–1393. DOI: 10.1109/CVPR.2014.180. URL: https://doi.org/10.1109/CVPR.2014.180.

[277] Tongzhou Wang and Phillip Isola. “Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 9929–9939. URL: http://proceedings.mlr.press/v119/wang20k.html.

[278] Xiaolong Wang and Abhinav Gupta. “Unsupervised Learning of Visual Representations Using Videos”. In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 2015, pp. 2794–2802. DOI: 10.1109/ICCV.2015.320. URL: https://doi.org/10.1109/ICCV.2015.320.

[279] Xiaosong Wang et al. “ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*.

IEEE Computer Society, 2017, pp. 3462–3471. DOI: 10.1109/CVPR.2017.369. URL: https://doi.org/10.1109/CVPR.2017.369.

[280] Xudong Wang et al. *Cut and Learn for Unsupervised Object Detection and Instance Segmentation*. Jan. 26, 2023. arXiv: 2301.11320 [cs]. preprint.

[281] Yangtao Wang et al. *TokenCut: Segmenting Objects in Images and Videos with Self-supervised Transformer and Normalized Cut*. Dec. 5, 2023. arXiv: 2209.00383 [cs, stat]. preprint.

[282] Ziyu Wang, Mike Zheng Shou, and Mengmi Zhang. *Object-Centric Learning with Cyclic Walks between Parts and Whole*. Feb. 15, 2023. arXiv: 2302.08023 [cs]. preprint.

[283] Nicholas Watters et al. *Spatial Broadcast Decoder: A Simple Architecture for Learning Disentangled Representations in VAEs*. Aug. 14, 2019. arXiv: 1901.07017 [cs, stat]. preprint.

[284] Jason Wei et al. "Emergent Abilities of Large Language Models". 2022. arXiv: 2206.07682.

[285] Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul. "Distance Metric Learning for Large Margin Nearest Neighbor Classification". In: *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*. 2005, pp. 1473–1480. URL: https://proceedings.neurips.cc/paper/2005/hash/a7f592cef8b130a6967a90617db5681b-Abstract.html.

[286] Philippe Weinzaepfel et al. *CroCo: Self-Supervised Pre-training for 3D Vision Tasks by Cross-View Completion*. Oct. 19, 2022. arXiv: 2210.10716 [cs]. preprint.

[287] Tom J. Wills et al. "Attractor Dynamics in the Hippocampal Representation of the Local Environment". In: *Science* 308.5723 (2005), pp. 873–876.

[288] Yi-Fu Wu et al. "Inverted-Attention Transformers Can Learn Object Representations: Insights from Slot Attention". In: *Causal Representation Learning Workshop at NeurIPS 2023*. 2023.

[289] Zhirong Wu et al. "Unsupervised Feature Learning via Non-Parametric Instance Discrimination". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018, pp. 3733–3742. DOI: 10.1109/CVPR.2018.00393. URL: http://openaccess.thecvf.com/content%5C_cvpr%5C_2018/html/Wu%5C_Unsupervised%5C_Feature%5C_Learning%5C_CVPR%5C_2018%5C_paper.html.

[290] Ziyi Wu et al. *SlotDiffusion: Object-Centric Generative Modeling with Diffusion Models*. Sept. 21, 2023. arXiv: 2305.11281 [cs]. preprint.

[291] Ziyi Wu et al. *SlotFormer: Unsupervised Visual Dynamics Simulation with Object-Centric Models*. Jan. 20, 2023. arXiv: 2210.05861 [cs]. preprint.

[292] Tete Xiao et al. "What Should Not Be Contrastive in Contrastive Learning". Aug. 12, 2020. arXiv: 2008.05659.

[293] Jiahao Xie et al. "Delving into Inter-Image Invariance for Unsupervised Visual Representations". Aug. 26, 2020. arXiv: 2008.11702.

[294] Saining Xie et al. "Rethinking Spatiotemporal Feature Learning: Speed-accuracy Trade-Offs in Video Classification". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 305–321. arXiv: 1712.04851.

[295] Yuwen Xiong, Mengye Ren, and Raquel Urtasun. "LoCo: Local Contrastive Representation Learning". Aug. 4, 2020. arXiv: 2008.01342.

[296] Jiarui Xu et al. *GroupViT: Semantic Segmentation Emerges from Text Supervision*. July 18, 2022. arXiv: 2202.11094 [cs]. preprint.

[297] Keyulu Xu et al. "How Powerful are Graph Neural Networks?" In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=ryGs6iA5Km.

[298] Guanglei Yang et al. "Uncertainty-Aware Contrastive Distillation for Incremental Semantic Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.2 (2023), pp. 2567–2581. DOI: 10.1109/TPAMI.2022.3163806.

[299] Zhilin Yang et al. "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach et al. 2019, pp. 5754–5764. URL: https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html.

[300] Mang Ye et al. "Unsupervised Embedding Learning via Invariant and Spreading Instance Feature". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 6210–6219. DOI: 10.1109/CVPR.2019.00637. URL: http://openaccess.thecvf.com/content%5C_CVPR%

5C_2019/html/Ye%5C_Unsupervised%5C_Embedding%5C_Learning%5C_via%5C_Invariant%5C_and%5C_Spreading%5C_Instance%5C_Feature%5C_CVPR%5C_2019%5C_paper.html.

[301] Kexin Yi et al. "CLEVRER: Collision Events for Video Representation and Reasoning". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=HkxYzANYDB.

[302] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. Ed. by Zoubin Ghahramani et al. 2014, pp. 3320–3328. URL: https://proceedings.neurips.cc/paper/2014/hash/375c71349b295fbe2dcdca9206f20a06-Abstract.html.

[303] Hong-Xing Yu, Leonidas J. Guibas, and Jiajun Wu. *Unsupervised Discovery of Object Radiance Fields*. Mar. 16, 2022. DOI: 10.48550/arXiv.2107.07905. arXiv: 2107.07905 [cs]. preprint.

[304] Jiahui Yu et al. "Vector-Quantized Image Modeling with Improved VQGAN". In: *International Conference on Learning Representations* (2021).

[305] Jure Zbontar et al. "Barlow Twins: Self-supervised Learning via Redundancy Reduction". In: *International Conference on Machine Learning*. PMLR, 2021, pp. 12310–12320.

[306] Matthew D. Zeiler and Rob Fergus. "Visualizing and Understanding Convolutional Networks". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer, 2014, pp. 818–833.

[307] Xiaohua Zhai et al. "Sigmoid Loss for Language Image Pre-Training". 2023. arXiv: 2303.15343.

[308] Chi Zhang et al. "ACRE: Abstract Causal REasoning beyond Covariation". 2021. arXiv: 2103.14232.

[309] Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. *Is an Object-Centric Video Representation Beneficial for Transfer?* July 20, 2022. arXiv: 2207.10075 [cs]. preprint.

[310] Richard Zhang, Phillip Isola, and Alexei A. Efros. "Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 645–654. DOI: 10.1109/CVPR.2017.76. URL: https://doi.org/10.1109/CVPR.2017.76.

[311] Ruixiang Zhang et al. *Robust and Controllable Object-Centric Learning through Energy-based Models*. Oct. 11, 2022. arXiv: 2210.05519 [cs]. preprint.

[312] Nanxuan Zhao et al. "What Makes Instance Discrimination Good for Transfer Learning?" June 11, 2020. arXiv: 2006.06606.

[313] Jinghao Zhou et al. *iBOT: Image BERT Pre-Training with Online Tokenizer*. Jan. 27, 2022. DOI: 10.48550/arXiv.2111.07832. arXiv: 2111.07832 [cs]. preprint.

[314] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. "Local Aggregation for Unsupervised Learning of Visual Embeddings". In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 6001–6011. DOI: 10.1109/ICCV.2019.00610. URL: https://doi.org/10.1109/ICCV.2019.00610.