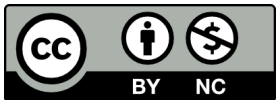
Efficient Self-Supervised Visual Representation Learning via Sparsity

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Sepehr Sameni
von Iran

Leiter der Arbeit:
Prof. Dr. Paolo Favaro
Institut für Informatik



Efficient Self-Supervised Visual Representation Learning via Sparsity

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Sepehr Sameni
von Iran

Leiter der Arbeit:
Prof. Dr. Paolo Favaro
Institut für Informatik

Von der Philosophisch-naturwissenschaftlichen Fakultät angenommen.

Bern, 06.09.2024

Der Dekan:
Prof. Dr. Jean-Louis Reymond

Abstract

Large collections of labeled data have greatly improved the performance of Deep Neural Networks in computer vision tasks. However, the vast majority of visual data generated daily remains unlabeled, limiting the potential of supervised learning paradigms. This thesis explores novel techniques to guide deep models towards learning generalizable visual patterns without human supervision, with a particular focus on leveraging sparsity as a key principle.

Our primary tool in this endeavor is the design of Self-Supervised Learning (SSL) tasks that do not require manual labeling. Beyond enabling learning from vast amounts of unlabeled data, we demonstrate how sparsity-based self-supervision can capture relevant patterns often overlooked by traditional supervised approaches. We design learning tasks that extract rich representations from various visual modalities: shape information from images, temporal dynamics from videos, and multimodal understanding from vision-language data.

A common thread running through our work is the strategic application of sparsity. In contrastive learning, we show how token sparsity can enhance both computational efficiency and representation quality. For video analysis, we leverage spatio-temporal sparsity to enable efficient and scalable representation learning. In generative tasks, we demonstrate how sparse conditioning can tackle complex problems like video prediction while implicitly modeling world dynamics.

Notably, our task designs follow a unifying principle: the recognition and manipulation of sparse patterns in data. The strong performance of the learned representations on downstream vision tasks such as image classification, video understanding, and multimodal reasoning validates this approach.

By consistently demonstrating that thoughtful application of sparsity can not only reduce computational demands but often improve the quality and generalizability of learned representations, this work lays a foundation for more efficient, scalable, and effective visual understanding systems. Our contributions pave the way for artificial systems with visual perception and reasoning capabilities that can better leverage the vast amounts of unlabeled visual data surrounding us.

Acknowledgments

In the grand tapestry of life, my PhD journey stands out as a vibrant, chaotic, and transformative thread. It's a tale that begins with a twist of geopolitical irony. To the despots of Iran, whose oppression inadvertently set me on this path: your attempts to stifle have only fanned the flames of curiosity. Who knew tyranny could be such an excellent career counselor?

To Professor Paolo Favaro, my advisor and academic shepherd, I owe a debt of gratitude that words can scarcely capture. His razor-sharp intellect and ability to generate innovative ideas have been a constant source of inspiration. Professor Favaro's unwavering commitment to pushing the boundaries of unsupervised learning challenged me to think deeper, profoundly reshaping my approach to research and problem-solving. His leadership fostered a friendly and collaborative environment in the lab, nurturing both individual growth and team synergy. Professor Favaro's purposeful guidance has been the compass for my doctoral journey, consistently pushing me beyond perceived limitations.

I am particularly thankful for the freedom Professor Favaro granted me to explore my own ideas, even when they led to dead ends. These experiences of trial and error were invaluable, teaching me resilience and the art of learning from failure. His trust in allowing me to teach a seminar offered a taste of academia beyond research, providing crucial insights for my future career decisions. For the robust foundation, the freedom to explore, and the constant push towards excellence, I am profoundly grateful. Professor Favaro's impact on my development will undoubtedly extend far beyond this thesis.

As this journey nears its end, I extend my thanks to Dr. Alexandre Alahi, my external examiner, for agreeing to scrutinize years of work in a matter of hours. May his energy be boundless as his patience.

Throughout this adventure, I've been blessed with a support system that spans continents. Nina and her family were the sunshine piercing through the often gloomy Swiss skies. Without Nina, this thesis might have veered into science fiction. Her family's warmth made Bern feel like home, even when my actual home felt galaxies away.

Speaking of home, my own family's love transcended borders and time zones, becoming a constant source of strength. To my sweet niece, whose hugs I had to ration like precious data points: one day, I'll explain why Uncle kept disappearing into his computer screen. Her innocent wonder at the world serves as a reminder of why we pursue knowledge.

In the lab, I found a second family. Givi, my academic big brother, showed me the ropes when I was just a bundle of nerves and excitement. Our COVID-era adventures proved that even in isolation, science (and friendship) finds a way. Simon, my unofficial co-supervisor, provided invaluable guidance. Thanks to him, I experienced an incredible internship at Adobe, an amazing journey that was as enriching as this PhD. His mentorship has been instrumental in shaping my research path.

Aram, my companion in contemplative walks and lunch break philosophizing, made our discussions often more nourishing than the food itself. Our shared journey through the highs and lows of PhD life created a bond that will last beyond these academic years.

The PhD student brigade (Adam, Abdelhak, Llukman, Alp, Adrian, Viktor, Hamadi, Luca, and Kaining) transformed this journey from potential solitary confinement into a chaotic, brilliant family reunion. Each of you added your unique flavor to this experience, making the lab a home.

Outside the academic bubble, Amin's friendship provided the perfect antidote to the rigid world of research. His spontaneity was a constant reminder that life exists and thrives outside the lab.

In those countless hours waiting for training runs to finish, YouTube became an unlikely ally, providing entertainment and distraction when I needed it most. Who knew that Starcraft matches and documentaries could be so crucial to the PhD process?

A special note of thanks goes to Dr. Mohammad Amin Sadeghi, my master's advisor. His unwavering belief in my abilities, even when I doubted myself, and his push towards excellence laid the foundation for this doctoral journey. His mentorship during my master's studies was instrumental in shaping my research aspirations and preparing me for the challenges of a PhD.

Lastly, I owe an immense debt of gratitude to my trio of best friends from high school: Foroot, Haji, and Shahrokhi, now scattered across the globe. Our weekly digital gatherings were my sanity's life support. Despite two of you embarking on your own PhD journeys, you've all patiently listened to more complaints about neural networks than anyone should endure. Your unwavering support and friendship have been the bedrock of this adventure.

I would be remiss not to acknowledge the field of self-supervised learning and the chatbots it has produced. These technological marvels not only advanced my research but also helped me articulate my thoughts, giving shape to this poetic acknowledgment.

As I close this chapter, I realize that the true value of a PhD lies not just in the contributions to science, but in the stories we collect, the friendships we forge, and the person we become. This thesis is more than a collection of papers; it's a testament to the power of human connections, the resilience of the spirit, and the unexpected joy found in pursuing knowledge.

To everyone mentioned here and the countless others who've touched this journey: thank you for being part of this beautiful, chaotic, and transformative adventure. You've all contributed to the researcher and the person I've become.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computer vision, a field at the forefront of artificial intelligence, has undergone a remarkable transformation in recent years. The advent of self-supervised learning (SSL) techniques has ushered in a new era, fundamentally changing how we approach the challenge of teaching machines to understand and interpret visual data. This paradigm shift moves beyond the constraints of traditional supervised learning, opening up new possibilities for harnessing the vast amounts of unlabeled visual data available in the world [33, 136].

While SSL has shown great promise in developing rich, transferable visual representations, it also faces significant challenges. The computational demands of processing large-scale visual datasets, the need for more efficient and scalable algorithms, and the quest for high-quality, generalizable features are at the forefront of current research [34, 100]. These challenges are particularly acute as we move towards processing higher resolution images, longer video sequences, and complex multimodal data.

This thesis explores a promising approach to address these challenges: sparsity. By strategically reducing the computational load through sparse processing of visual information, we aim to enhance both the efficiency and effectiveness of SSL algorithms. Our work investigates how sparsity can be leveraged across various aspects of SSL, from contrastive learning and video understanding to generative models and multimodal learning.

Through a series of interconnected studies, we demonstrate that intelligent application of sparsity can not only mitigate the computational burdens of SSL but also improve the quality of learned representations. Our contributions span from novel loss functions that induce shape bias in sparse SSL, to efficient post-training methods for video understanding, to sparse conditioning in generative video prediction models. We also explore how sparsity can be applied to knowledge distillation from foundation models, enhancing vision-language representations while maintaining computational efficiency.

By addressing the core challenges of SSL through the lens of sparsity, this thesis aims to contribute to the development of more efficient, scalable, and effective self-supervised learning methods for visual understanding. Our work not only advances the state-of-the-art in specific

visual tasks but also provides insights into the broader potential of sparsity as a key principle in designing the next generation of SSL algorithms.

1.1 Self-Supervised Learning in Computer Vision

1.1.1 Fundamentals of SSL

Self-supervised learning (SSL) has emerged as a powerful paradigm in computer vision, addressing many limitations of traditional supervised learning approaches. At its core, SSL leverages the inherent structure and patterns within unlabeled data to learn meaningful representations without the need for explicit human annotation [61, 136].

The fundamental principle of SSL in computer vision is the design of pretext tasks. These tasks force models to learn useful features by solving carefully crafted problems that can be automatically generated from the data itself. Common pretext tasks include predicting the relative position of image patches [61], solving jigsaw puzzles of scrambled image parts [192], colorizing grayscale images [292], and reconstructing corrupted or masked inputs [110]. By solving these tasks, models develop a rich understanding of visual concepts that can be transferred to a wide array of downstream tasks, often surpassing the performance of models trained with traditional supervised methods [38].

The advantages of SSL over supervised learning are significant. SSL can utilize vast amounts of unlabeled data, which is abundantly available and easier to collect compared to labeled datasets [34]. By learning from diverse, unlabeled data, SSL models can potentially avoid biases introduced by human annotation [56]. Furthermore, features learned through SSL often demonstrate better transfer to various downstream tasks [38], enhancing their generalizability.

1.1.2 Recent Advancements

Recent years have witnessed remarkable advancements in SSL for computer vision, driven by several key factors. The availability of massive unlabeled datasets has enabled SSL models to learn from a diverse range of visual concepts [100]. The introduction of Vision Transformers (ViTs) [67] has provided a powerful backbone for SSL models, capable of capturing long-range dependencies in visual data. Additionally, novel contrastive and non-contrastive learning approaches have enhanced the quality of learned representations [33, 38, 102].

These advancements have led to impressive results across various domains. In image understanding, SSL models have achieved state-of-the-art performance in tasks such as image classification, object detection, and semantic segmentation [34]. Extensions of SSL techniques to video data have enabled improved action recognition and temporal understanding [244]. In the realm of multimodal learning, SSL ideas has been successfully applied to learn joint representations of images and text, leading to powerful vision-language models [209]. These

models often leverage weakly supervised learning approaches, utilizing large-scale paired image-text data to learn robust visual-linguistic associations without the need for explicit task-specific annotations.

As SSL continues to evolve, it promises to further bridge the gap between human and machine perception, enabling more sophisticated and generalizable visual understanding systems. However, realizing this potential requires addressing several key challenges, which we will explore in the next section.

1.2 Challenges in Self-Supervised Learning

Despite the promising advancements in self-supervised learning (SSL), several significant challenges remain that hinder its full potential in computer vision applications. These challenges span various aspects of SSL, from computational constraints to the quality and generalizability of learned representations.

One of the most pressing issues is computational efficiency. As SSL models and datasets continue to grow in size and complexity, the computational demands become increasingly prohibitive [226]. This challenge extends beyond the financial cost of training to include the environmental impact of AI research and deployment. The need for efficient learning algorithms and architectures that can handle large-scale visual data without excessive computational overhead is paramount.

Closely related to computational efficiency is the challenge of scalability. As we move towards processing higher resolution images, longer video sequences, and complex multimodal data, the scalability of SSL approaches becomes crucial. Methods that work well on small-scale problems may not necessarily translate to larger, more complex datasets, presenting challenges in both algorithm design and resource allocation [100]. Developing SSL techniques that can efficiently scale to handle these more demanding visual tasks is essential for the continued advancement of the field.

Another critical challenge lies in the quality and generalizability of the learned features. While SSL has shown promise in many areas, consistently exceeding the quality of features learned through supervised methods remains an ongoing challenge [38]. This requires developing sophisticated learning objectives and architectures that can extract rich, hierarchical representations from unlabeled data, comparable or superior to those learned from carefully curated labeled datasets. Moreover, ensuring that these learned representations generalize well across diverse downstream tasks and domains is crucial for the practical applicability of SSL models.

The design of effective pretext tasks presents another significant challenge. While the introduction touched on the principle of pretext tasks, the challenge lies in designing tasks that lead to truly meaningful and transferable representations. This remains as much an art as a science, with the choice of pretext task significantly impacting the quality and applicability

of the learned features [136]. Creating tasks that induce the learning of semantically rich and task-agnostic representations is an ongoing area of research.

Lastly, the challenge of bridging the gap between pre-training and fine-tuning persists. While SSL has shown remarkable transfer learning capabilities, optimizing the process of adapting pre-trained representations to specific downstream tasks, especially in low-data regimes, remains an important area of investigation. This includes developing methods for efficient fine-tuning and exploring few-shot learning scenarios where SSL can provide a strong foundation.

Addressing these challenges is crucial for realizing the full potential of SSL in computer vision. The next section will introduce sparsity as a key concept in our approach to tackling these challenges, setting the stage for the specific contributions of this thesis.

1.3 Sparsity: A Key to Efficient SSL

In light of the challenges facing self-supervised learning, this thesis explores a promising approach to address them: sparsity. While sparsity is not a new concept in machine learning, its application to SSL in visual domains presents unique opportunities to enhance both the efficiency and effectiveness of learning algorithms.

In the context of our work, sparsity refers to the strategic reduction of computational load by processing only a subset of available information. This can manifest in various forms:

- **Input Sparsity:** This involves processing only a portion of input tokens or frames. In the realm of computer vision, this could mean selecting some patches from images or random frames from video sequences. By focusing computational resources on a small subset of the input, we can significantly reduce processing time and memory requirements without substantially compromising the quality of learned representations.

- **Loss Sparsity:** This approach focuses the learning objective on a subset of outputs or features. By carefully designing loss functions that prioritize the most critical aspects of the learning task, we can guide the model to learn more efficiently and potentially capture more meaningful representations.

The application of sparsity to SSL offers several potential benefits in addressing the challenges outlined in the previous section:

- **Computational Efficiency:** By processing only a subset of the input or focusing on key aspects of the learning objective, sparse SSL methods can significantly reduce the computational demands of training and inference. This not only makes SSL more practical to deploy at scale but also aligns with growing concerns about the environmental impact of AI research and deployment [226].

- **Scalability:** Sparse approaches enable the processing of higher resolution inputs and longer sequences, which is critical for tasks involving high-quality images or long videos. This scalability is crucial as we move towards more complex visual understanding tasks that require finer-grained analysis or longer temporal contexts.
- **Feature Quality:** Counterintuitively, strategic sparsity can lead to improved feature quality. By forcing models to focus on the most informative parts of the input, we can potentially learn more robust and generalizable representations. This aligns with cognitive science research suggesting that human visual processing often relies on sparse, key features rather than exhaustive analysis of every detail [195].
- **Task Design:** Sparsity opens up new possibilities in designing pretext tasks for SSL. By focusing on sparse subsets of the input or output space, we can create more challenging and informative tasks that encourage the model to learn rich, transferable representations.
- **Transfer Learning:** Sparse representations can be particularly beneficial in transfer learning scenarios. They can capture essential features that are more likely to be relevant across different tasks and domains, potentially improving the efficiency of fine-tuning and few-shot learning.

In the subsequent chapters, we will explore how these principles of sparsity can be applied across various aspects of SSL in computer vision. From enhancing contrastive learning methods to improving video understanding and generative models, we will demonstrate that sparsity is not just a tool for efficiency, but a key to unlocking more effective and generalizable self-supervised learning algorithms.

1.4 Thesis Contributions

This thesis explores and demonstrates how sparsity can be leveraged to enhance SSL models and representations in visual domains. Through five interconnected studies, we address the challenges outlined earlier and make the following key contributions:

1.4.1 DILEMMA: Sparse Contrastive Learning with Shape Bias

In Chapter 3, we investigate the impact of token sparsity [3, 110] on contrastive learning methods. We uncover potential pitfalls in naive sparse approaches and propose DILEMMA (Detection of Incorrect Location EMbeddings with MAsked inputs), a novel loss function that induces a shape bias [94]. This method not only reduces computational load through input sparsity but also improves the quality of learned representations, particularly in shape-based tasks. DILEMMA demonstrates how thoughtful application of sparsity can enhance both efficiency and feature quality in SSL.

1.4.2 SCALE: Sparse Aggregation for Video Understanding

Chapter 4 introduces SCALE (Spatio-temporal Crop Aggregation for video representation LEarning), a post-training method that leverages sparsity to improve pretrained features and extend their temporal range in video understanding tasks. By sparsely sampling video clips and applying a self-supervised objective of masked clip feature prediction, SCALE enables the use of higher resolution inputs and enhances long-term video understanding capabilities. This work showcases how sparsity can be used to efficiently improve existing models and scale to longer temporal ranges.

1.4.3 RIVER: Sparse Conditioning for Video Prediction

In Chapter 5, we extend the concept of sparsity to generative models, specifically video prediction. RIVER (Random frame conditioned flow Integration for VidEo pRediction) introduces sparse conditioning on past frames, making long-term video prediction computationally feasible while maintaining high-quality outputs. By applying sparsity to conditional frames in generative models, RIVER demonstrates the potential of sparse techniques beyond discriminative tasks. This work highlights the importance of exploring sparsity in the context of world models [104], which are implicit representation learners. By showing that sparse conditioning can be effective in video prediction, RIVER opens new avenues for efficient generative modeling in SSL, potentially bridging the gap between discriminative and generative approaches.

1.4.4 ViDROP: Efficient Video Representation Learning

Chapter 6 introduces ViDROP (Video Dense Representation through Omissive Processing), addressing limitations in current video representation models. Unlike existing approaches that rely on reconstruction-based methods requiring expensive fine-tuning [18, 84, 244, 256], ViDROP combines masked token reconstruction with token dropping in an encoder-only model. This novel approach achieves efficient and effective representation learning for videos, eliminating the need for costly fine-tuning typically associated with sparse encoder / dense decoder models. By enabling direct linear probing on learned representations, ViDROP significantly reduces the gap between pretrained features and fine-tuned ones. This innovation demonstrates how sparsity can overcome key challenges in video SSL, paving the way for more efficient and adaptable video understanding models.

1.4.5 SF-CLIP: Sparse Distillation for Multimodal Learning

Finally, in Chapter 7, we apply sparsity in the context of multimodal learning. SF-CLIP (Solid Foundation CLIP) uses sparse distillation from foundation models to enhance vision-language representations. This method improves zero-shot learning and multilingual performance while maintaining computational efficiency. SF-CLIP demonstrates how sparsity can be

applied to knowledge distillation from foundation models [20] without sacrificing training efficiency, merging strong spatial representations with the advanced language capabilities of Large Language Models (LLMs).

1.4.6 Overarching Contributions

Collectively, these works make several overarching contributions to the field of SSL:

1. We demonstrate that sparsity, when thoughtfully applied, can significantly reduce computational demands without compromising - and often improving - the quality of learned representations.
2. Our work shows how sparsity can be leveraged across different modalities (images, videos) and learning paradigms (contrastive learning, generative models, and multimodal learning).
3. We provide novel insights into the design of sparse SSL algorithms, offering principles that can guide future research in this area.
4. Our contributions open new avenues for scaling SSL to higher resolution inputs, longer temporal sequences, and more complex multimodal scenarios.

By addressing core challenges in SSL through the lens of sparsity, this thesis contributes to the development of more efficient, scalable, and effective self-supervised learning methods for visual understanding. Our work not only advances the state-of-the-art in specific visual tasks but also provides insights into the broader potential of sparsity as a key principle in designing the next generation of SSL algorithms.

The list of research works associated with each chapter:

1. Chapter 3 - "Representation learning by detecting incorrect location embeddings" [220], in AAAI 2023.
2. Chapter 4 - "Spatio-Temporal Crop Aggregation for Video Representation Learning" [221], in ICCV 2023.
3. Chapter 5 - "Efficient Video Prediction via Sparsely Conditioned Flow Matching" [53], in ICCV 2023.
4. Chapter 6 - "ViDROP: Video Dense Representation through Omissive Processing", manuscript in preparation.
5. Chapter 7 - "Building Vision-Language Models on Solid Foundations with Masked Distillation" [222], in CVPR 2024.

Chapter 2

Background

Self-supervised learning (SSL) has emerged as a powerful paradigm in unsupervised learning, utilizing pretext tasks created from unlabeled data to provide supervision signals. This approach offers advantages over traditional supervised learning, such as reduced reliance on expensive labeled data and improved generalization. In natural language processing (NLP), the success of transformer architectures [249] can be attributed to extensive self-supervised pre-training, with models like BERT [58] and the GPT series [24, 207, 208] demonstrating significant improvements through the use of vast amounts of unlabeled data.

In computer vision, SSL is driving similar advancements to those seen in NLP. A key breakthrough has been the introduction of Vision Transformers (ViTs) [67], which adapt the transformer architecture [249] for vision tasks. Models like VideoMAEv2 [256] and DINOv2 [196] exemplify the potential of scaling up both model size and dataset volume in SSL for vision tasks. These models' impressive performance and versatility make them strong candidates to be considered foundation models [20] - large-scale models trained on broad data that can be adapted to a wide range of downstream tasks. These developments underscore SSL's crucial role in enhancing the capabilities of large-scale vision models, mirroring the impact seen in NLP.

This chapter provides a comprehensive overview of SSL, focusing on its applications to images and videos, and exploring multimodal weakly supervised learning. We begin by discussing SSL methods for images, covering three main approaches: proxy tasks (self-created tasks that induce useful representations), consistency-based methods, and reconstruction-based approaches. Following that, we examine SSL for videos, addressing the unique challenges and opportunities presented by temporal data, and highlighting video generation techniques. Finally, we explore the intersection of SSL with multimodal learning, tracing the progression from purely unsupervised methods to weakly supervised approaches. This section culminates in a discussion of influential models like CLIP [209], which leverage large-scale image-text pairs to learn powerful visual representations, bridging the gap between vision and language understanding.

2.1 Self-Supervised Learning for Image Representations

2.1.1 Proxy Tasks

Proxy tasks in self-supervised learning for image representations provide supervision signals without requiring labeled data, inducing the learning of useful features that transfer well to downstream vision tasks. Classic examples include classifying image patch locations [61, 192], where the model learns spatial relationships by predicting the relative positions of image patches, thereby developing an understanding of image composition and structure. Another common task is reconstructing color channels [292], which challenges the model to predict the color of grayscale images, encouraging recognition of objects and textures based on shape and pattern. Additionally, recognizing various image transformations, such as rotations [95], helps models learn rotation-sensitive features. This task encourages the model to understand concepts like orientation, the typical positioning of objects (e.g., sky at the top), and shape characteristics, rather than relying solely on low-level textures. Spotting specific artifacts [132] further develops the model's sensitivity to fine-grained image details. While not directly related to end-goal vision tasks, these proxy tasks encourage models to learn generalizable features valuable in a wide range of computer vision applications.

Building upon these classic approaches, recent advancements in proxy tasks have introduced more sophisticated modifications of image content and positions. These newer tasks leverage the capabilities of Vision Transformers (ViTs) [67], particularly their position encoding, which enables complex spatial manipulations of image data. This evolution allows for more challenging and informative pretext tasks, potentially leading to richer learned representations. For instance, Corrupted Image Modeling (CIM) [80], drawing inspiration from the ELECTRA model [47] in NLP, tasks the model with detecting corrupted image tokens produced by models like BEiT [16]. This approach encourages the model to develop a nuanced understanding of image composition and content. Another innovative method, MP3 [288], extends the concept of jigsaw puzzles [192] to predict the positions of all tokens in an image, fully utilizing ViTs for enhanced representation learning. While not all new approaches yield significant improvements - for example, DABS [236] introduces patch misplacement detection without notably boosting performance - these explorations continue to push the boundaries of what can be learned through self-supervision in vision tasks.

2.1.2 Consistency-Based Methods

Consistency-based methods have become a cornerstone of self-supervised learning (SSL) in computer vision, aiming to learn representations that remain consistent across various transformations or views of the same data. At the heart of this approach lies contrastive learning [248], which has gained prominence through instance discrimination techniques [66,

272]. These methods learn to distinguish each training instance from others, even after applying data augmentations, laying the groundwork for more advanced approaches.

Building on this foundation, frameworks like CPC [248] and SimCLR [33] have further popularized contrastive learning by demonstrating its effectiveness in learning robust representations. These methods typically involve comparing positive pairs (augmented versions of the same image) against negative pairs (different images) in a high-dimensional feature space.

Subsequent research has introduced various enhancements to the basic contrastive paradigm. Momentum Contrast (MoCo) [38, 109] leverages a momentum-encoded key encoder to maintain a consistent dictionary of negative samples, improving training stability. BYOL [102] and SimSiam [37] take a different approach by eliminating the need for explicit negative pairs, instead focusing on avoiding representational collapse - a phenomenon where the model outputs trivial, non-informative representations. These methods employ techniques such as stop-gradient operations and predictor networks to maintain meaningful learning dynamics.

Clustering-based approaches like SwAV [28] extend the concept of positive samples beyond simple data augmentation by grouping similar features, enabling more flexible learning of visual concepts. NNCLR [71] further refines this idea by using nearest neighbors in feature space to generate positive samples, enhancing the diversity of comparisons during training.

As the field has advanced, researchers have adapted consistency-based methods for dense prediction tasks, which require detailed spatial understanding. Approaches such as VADER [193], DenseCL [259], ReSim [275], PixPro [278], DSC [159], and DRLoc [169] tailor contrastive pre-training strategies to focus on dense feature maps rather than global representations. These methods enable the learning of spatially-aware features crucial for tasks like semantic segmentation and object detection.

The advent of vision transformer architectures [67, 171] has spurred further innovations in consistency-based methods. Approaches like MoCov3 [39] and SwinSSL [277] adapt existing contrastive frameworks to leverage the unique properties of transformers, such as their ability to model long-range dependencies. Concurrently, new architectures tailored specifically for SSL, such as EsViT [155], have emerged, along with novel objectives like those introduced in iBOT [299].

2.1.3 Reconstruction-Based Methods

Reconstruction-based methods have evolved significantly in self-supervised learning for image representations. Traditional autoencoders [218] laid the groundwork by encoding input data into a latent space and then reconstructing the original input. Recent advancements have built upon this foundation, introducing more sophisticated approaches. Masked Autoencoders (MAE) [110], for instance, improve learning by regressing missing pixel patches within given contexts, forcing the model to understand the underlying structure of the image.

Beyond pixel-space reconstruction, models like AIM [74] and BEiT [16] perform infilling in a fixed latent space, capturing more semantic information crucial for downstream tasks.

Hybrid approaches, such as iBOT [299] and DINOv2 [196], further enhance this method by integrating features of both reconstruction (via self-distillation [13]) and consistency learning. These advancements have led to high-quality representations that excel in various tasks, from KNN classification [88] to unsupervised semantic segmentation [105], demonstrating the versatility and power of reconstruction-based SSL methods in computer vision.

2.1.4 Leveraging SSL Representations

Self-supervised learning (SSL) in computer vision has demonstrated its versatility through various applications, mirroring the success seen in natural language processing (NLP). Fine-tuning, a technique where pre-trained models are adapted to specific tasks with minimal labeled data, has shown remarkable effectiveness. This approach, analogous to BERT [58] in NLP, has proven valuable in vision tasks, particularly with newer methods like VideoMAEv2 [256]. However, while efficient in label usage, fine-tuning can be computationally intensive, especially for video data.

Linear probing offers a quicker assessment of learned representations' quality. This technique, exemplified by contrastive methods such as SimCLR [33] and the weakly supervised CLIP [209], is comparable to the robust performance observed with GPT-2 [208] in NLP. High linear probing accuracy indicates that a single model can be adapted for different tasks by simply changing the linear head, potentially saving significant costs associated with fine-tuning.

Few-shot learning, which has revolutionized NLP with models like GPT-3 [24], is still in its early stages in computer vision. This approach enables models to perform new tasks with very few labeled examples. Initial attempts in vision, such as those by Bai *et al*. [14] and Bar *et al*. [17], show promising potential. As research progresses, few-shot learning in computer vision is expected to bridge the gap between pre-training and downstream task performance, potentially reducing the need for large labeled datasets in specific applications.

2.1.5 Efficiency Considerations in SSL

As self-supervised learning models grow in size and complexity, addressing computational efficiency becomes crucial. FlashAttention [49, 50] has made significant strides in reducing memory costs for transformer architectures, addressing the $\mathcal{O}(N^2)$ memory problem of attention operations, where N is the number of image patches. However, the challenge of optimizing compute efficiency remains, and further memory reduction is still desirable, especially for processing videos that can contain thousands of tokens.

As an orthogonal solution, token dropping [3, 36, 73, 110] has emerged as a significant advancement in improving the training efficiency of vision transformers. This approach randomly omits tokens during training, reducing computational load without sacrificing performance. Initially explored through masking patches [16, 279], token dropping has proven more effective when combined with a shallow dense decoder using mask tokens.

2.2 Self-Supervised Learning for Video Representations

Self-supervised learning (SSL) for video data presents unique challenges and opportunities not encountered with still images. The temporal dimension of videos offers rich information for feature learning, but it also introduces complexity to learning algorithms and increases computational demands. This section explores various approaches to SSL in video, highlighting how researchers leverage the dynamic nature of video data to learn robust representations.

2.2.1 Proxy Tasks

One valuable strategy in self-supervised learning for video involves pretext tasks designed to capture both spatial and temporal aspects. For example, tasks that require the model to determine the correct order of shuffled video frames [52, 64, 131, 181] aid in understanding activity sequences within the videos. These tasks enable the model to learn temporal dependencies and spatial coherence, which are crucial for effective video representation learning.

2.2.2 Consistency-Based Methods

Consistency-based approaches in video SSL focus on identifying consistent features across different segments or views of the same video. These methods build upon techniques developed for image SSL but adapt them to the temporal nature of video data. A key strategy involves ensuring that models identify consistent features across different segments of the same video, such as clips captured at different times or under varied conditions [83, 212, 214]. This technique effectively captures invariant features across videos. To further enhance the learning of robust video representations, some approaches incorporate additional modalities like audio [183].

Advanced techniques in this domain combine contrastive methods with explicit temporal constraints [51], integrating consistency-based learning with pretext tasks [133] for more comprehensive feature learning. Some methods also incorporate optical flow [106] to capture motion information explicitly. These approaches typically focus on learning representations with limited temporal extent (few seconds), balancing the richness of temporal information with computational feasibility.

2.2.3 Reconstruction-Based Methods

Reconstruction-based approaches in video self-supervised learning leverage generative models to reconstruct segments or entire frames of videos, compelling the models to capture crucial visual and temporal details. Models like VideoMAE [244], SpatiotemporalMAE [84], and VideoMAEv2 [256] utilize masked input reconstruction methods, a technique popularized in image SSL [110], successfully translating it to video.

Other approaches formulate masked prediction tasks in the learned feature space [18] or in some fixed latent space [237], enabling the models to learn more abstract representations. Additionally, some methods consider directional predictions, such as forecasting future frames, often formulated via contrastive predictive coding [248]. This approach has been effectively applied to video data [96, 170, 173, 235, 271], emphasizing the temporal coherence and progression in video sequences.

2.2.4 Learning from Multiple Views

Learning from multiple views is a powerful paradigm in video SSL, exploiting the multi-dimensional nature of video data. This approach involves generating different perspectives or "view" of the same video content, often through space-time cropping or other transformations. Many self-supervised learning approaches leverage multiple views of the data, particularly in contrastive formulations [83, 201, 212] or predictive learning [214], where the goal is to achieve invariance to different views.

Beyond leveraging multiple views for SSL, some works propose general multi-view video models. These models capture and fuse features at different spatio-temporal resolutions [280], aggregate information over longer time spans [227, 254, 267, 268], or select important frames [99]. Other methods focus on learning from the relationships between two views. Some approaches predict overlap [293], relative distance [234], or engage in cross-view feature prediction [238, 286] and reconstruction [188]. These strategies help in capturing the coherence and consistency of features across various views, enhancing the robustness of the learned representations.

2.2.5 Addressing Computational Challenges

Processing videos for SSL is significantly more resource-intensive than processing images, especially when using Vision Transformers (ViT) [67]. Various techniques have been explored to mitigate these challenges. One method employs factorized attention [19], as utilized in SVT [212]; however, full spatio-temporal attention, as demonstrated in ViViT [6], generally delivers superior performance. Another strategy leverages pretrained encoders for images or short video clips to derive representations for longer video sequences [154].

Inspired by Masked Autoencoders (MAE) [110], using sparse input to the encoder paired with dense outputs from a relatively shallow decoder can also reduce computational demands. However, the cost associated with decoders remains significant at scale. Models like VideoMAEv2 [256], EVEREST [125], and CrossMAE [90] address this by introducing sparse reconstruction tokens to the decoder, further economizing on resources.

Addressing the data bottleneck presents another challenge, as video decoding is substantially more costly than image decoding. While tools like FFCV [150] and its SSL variant [21] are ineffective for video, wrappers such as decord [54] and Avion [296] offer some improve-

ments, though they remain relatively slow. Data remasking techniques [18, 84] provide a cost-effective way to alleviate data loading bottlenecks. Additionally, employing precomputed latent representations of videos, rather than raw pixels, can reduce data loading times [130, 266]. This approach may be integrated with learned data augmentations within this latent space [152], though it introduces higher costs at the inference stage due to the necessity of an expensive encoder to convert raw pixel videos into the latent representation, complicating the use of pretrained image models within the video domain.

2.2.6 Video Prediction

Video prediction models represent the pinnacle of unsupervised learning, often referred to as world modeling [104]. These models aim to generate realistic future frames of a video sequence based on past frames, requiring extensive knowledge about the physical world to predict future states accurately. This capability is analogous to the complex understanding demonstrated by advanced image generation models [157].

The field has evolved from traditional approaches using Recurrent Neural Networks (RNNs)[11] to more sophisticated methods. Variational techniques[141] have been widely adopted, often incorporating hierarchical structures to model longer sequences [153, 265]. To address the challenge of blurry predictions, researchers have explored hybrid approaches combining adversarial loss [98] with Variational Autoencoders [151]. The success of large language models [24] has inspired new directions, with autoregressive transformers [249] emerging as powerful alternatives to RNNs. To mitigate computational costs and scale to longer and higher-resolution videos, researchers have turned to vector-quantized codes [247]. These codes, obtained using techniques like VQGAN [75], are predicted either on a per-frame basis [103, 149, 210, 228] or for sets of frames [281], offering a more efficient representation of video content.

Score-based diffusion models [232], a class of generative models that gradually transform noise into structured data, have shown impressive results in image generation [59, 211]. Building on this success, researchers have extended these models to the domains of video generation [117] and prediction [107, 116, 119, 252, 282]. While unconditional diffusion models can approximate conditional distributions [117], directly modeling the conditional distribution has been shown to yield better performance [240]. MCVD [252] introduces an innovative masking approach to train a single model capable of generating past, future, or intermediate frames. This technique allows for longer sequence generation by applying a moving window during training, though it comes at the cost of increased computational requirements when conditioning on past frames. Flexible diffusion models, such as FDM [107], tackle this challenge by employing a per-frame UNet [217] architecture with attention mechanisms. This design allows the model to handle a variable number of input frames, enabling conditioning on frames from the distant past and predicting multiple future frames simultaneously. These advancements in diffusion

models for video prediction demonstrate their potential to capture complex spatio-temporal dynamics, positioning them as a promising approach for future research in this field.

2.3 Multimodal Weakly Supervised Learning

This section focuses on weakly supervised Vision-Language Models (VLMs), which have gained prominence due to their wide range of applications. These models, trained on large-scale paired image-text data without explicit labels, have demonstrated remarkable capabilities in tasks such as zero-shot classification [209], visual question answering (VQA) [229], captioning [158], text-to-image generation [216], and have even been integrated into Multimodal Large Language Models (MLLMs) [166]. The weak supervision paradigm allows these models to learn from vast amounts of naturally occurring data, potentially capturing rich and diverse visual-linguistic relationships.

2.3.1 Contrastive Learning in VLMs

The contrastive learning framework [248] has been a cornerstone in the development of VLMs. This approach utilizes dual encoders to map images and text into a shared embedding space, where the goal is to maximize the similarity between corresponding image-text pairs while minimizing it for non-corresponding pairs [209]. Models like CLIP and ALIGN [134] have leveraged this technique to achieve impressive zero-shot learning capabilities across various visual tasks. However, while effective at capturing high-level associations, this method often struggles to encode finer compositional details [287], limiting the models' ability to understand complex visual scenes and their textual descriptions.

2.3.2 Challenges in Spatial and Linguistic Understanding

A significant limitation of standard contrastive training in VLMs is the insufficient capture of compositional information, object attributes, and relations [223, 295]. This shortcoming arises partly from the nature of web-scraped image-text pairs, which often lack the depth required for understanding complex compositions [78]. The literature identifies this as a critical area for improvement in VLMs [165, 197, 242], as it impacts the models' ability to perform tasks requiring fine-grained understanding of visual scenes.

To address these challenges, researchers have explored various approaches. Data-level interventions, such as data augmentation and hard-negative mining, have shown promise. Techniques include modifying text descriptions by word swapping or replacement [69, 287] and enhancing captions using large language models (LLMs) [70, 78, 182]. While these methods have demonstrated improvements, they risk overfitting to specific textual modifications and may introduce hallucinations, a common problem with large language models [124].

Another line of research introduces additional learning objectives to improve VLMs. This includes incorporating self-supervision in the vision and text branches [63, 160]. Although effective in enhancing model performance, these methods significantly increase the computational overhead of training large-scale VLMs [284]. For example, training MaskCLIP [63] requires 1.75× more resources than vanilla CLIP, and training SLIP [186] takes 2.67× more.

2.3.3 Leveraging Foundational Models

Recent works have demonstrated the efficacy of leveraging vision foundational models - large-scale models pretrained on diverse visual tasks - to enhance VLMs. These foundational models, such as DINOv2 [196] and SAM [142], have shown robust feature understanding and the ability to capture rich spatial features. This insight has guided research towards incorporating these models into VLM architectures.

For instance, LiT [289] uses a frozen pretrained vision encoder instead of training one from scratch, while Three Towers [144] employs a frozen vision encoder to guide CLIP's vision encoder. SAM-CLIP [253] takes a different approach, starting from a pretrained SAM [142] model and fine-tuning it with both SAM and CLIP objectives via another larger pretrained CLIP model. Despite the success of using pretrained vision models, the use of large language models (LLMs) for VLM representation learning is still under-explored. However, LLMs have shown promise when used as text rewriters in text-to-image generation models like DALLE-3 [230], demonstrating improved results.

2.3.4 Multilinguality

As VLMs find applications in increasingly global contexts, the ability to understand and generate content in multiple languages has become crucial. There are generally two approaches to training multilingual CLIP models, each with its own trade-offs between performance and computational efficiency.

The first method involves training the model directly on multilingual data. This approach is exemplified by mSigLIP [290], which is trained on WebLI [35], and OpenCLIP [41], trained on LAION5B [224]. By exposing the model to a wide range of languages during training, this method leads to more naturally generalized multilingual capabilities. However, it requires immense computational resources to handle such large and diverse datasets.

An alternative strategy to mitigate the high costs of direct multilingual training is to align CLIP's English text encoder with a pre-trained multilingual text encoder using parallel sentences [27, 40, 65]. This method is more resource-efficient, leveraging existing models and data. However, its effectiveness depends on the quality of the parallel sentences and the alignment process, which can vary based on the languages involved and the quality of the machine translation system used.

Chapter 3

DILEMMA: Representation Learning by Detecting Incorrect Location Embeddings

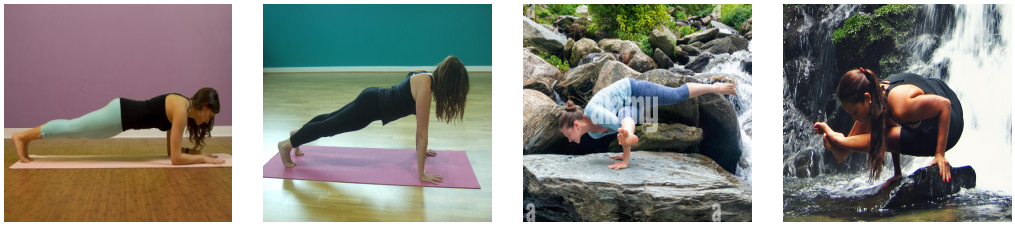
Material in this chapter is based on: Sameni, Sepehr, Simon Jenni, and Paolo Favaro. "Representation learning by detecting incorrect location embeddings." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 8, pp. 9704-9713. 2023. https://doi.org/10.1609/aaai.v37i8.26160

As we explore the role of sparsity in self-supervised learning (SSL), a critical challenge emerges: how can we leverage token sparsity in Vision Transformers (ViTs) to not only improve computational efficiency but also enhance the quality of learned representations? This chapter introduces DILEMMA (Detection of Incorrect Location EMbeddings with MAsked inputs), a novel approach that addresses this challenge while demonstrating how strategic sparsity can enhance model capabilities, particularly in shape discrimination.

Recent advancements in SSL have shown that pre-training with unlabeled data can surpass supervised pre-training on various downstream tasks [109]. However, the generalization ability of these representations, especially to shape-based tasks, remains an area ripe for improvement. Our work builds upon the insight that representations with a shape bias tend to generalize better to tasks such as object classification, detection, and segmentation [94, 239].

To illustrate this point, consider the transfer learning to a shape-based task like pose classification in the Yoga_{82} dataset [250] (Fig. 3.1). This example motivated our investigation into whether incorporating a shape-sensitive regularization loss into state-of-the-art SSL methods could lead to enhanced representation learning.

DILEMMA introduces two key components to foster shape discrimination in image representations. First, a binary classification loss for detecting correct/incorrect positions of



Dolphin plank | Plank | Sage Koundinya | Eight angle

Figure 3.1: **Difficulty of classifying Yoga$_{82}$ images.** As these examples show, texture alone is not sufficiently indicative of the pose. Thus, models that perform well in this task may demonstrate a strong shape discriminability.

object parts. Second, the utilization of randomized input sparsity with varying sparsity ratios, ensuring diverse subsets of object parts contribute to the overall image representation. These components draw inspiration from various SSL methods, including concept prediction [61], jigsaw puzzle approaches [192], and token replacement detection used in natural language processing [47]. Crucially, our implementation of input sparsity aligns with the broader theme of this thesis, building upon techniques used in models like VATT [3] and MAE [110] to reduce the computational workload of training with ViTs [67].

As illustrated in Fig. 3.2, our method operates by dividing an image into a grid of tiles, mapping them to tokens, and combining them with positional embeddings. We then corrupt the positional embeddings of a fraction of the tokens before feeding them to a ViT. The DILEMMA loss classifies tokens into those with correct and incorrect positional embeddings. Importantly, we implement input sparsification by discarding a randomized percentage of tokens, with the sparsity ratio varying for each input. This dynamic sparsity approach helps to close the gap between training and test data distributions.

We also employ a teacher-student architecture, reminiscent of MoCoV3 [39]. The student network receives sparsified input with varying sparsity ratios, while the teacher network processes all tiles. This approach not only reduces storage and computing resources but also provides a more robust reference for the student network across different sparsity patterns.

Our contributions can be summarized as follows:

- A novel SSL regularization loss that enhances the shape discriminability of image representations.
- Dynamic input sparsification and a teacher-student architecture to reduce memory usage, bridge training-test data gaps, and accelerate training.
- Demonstrable performance improvements for established SSL methods including MoCoV3 [39], SimCLR [33], DINO [29], and MAE [110] under equivalent computational budgets.

By focusing on shape discriminability and leveraging dynamic sparsity, DILEMMA not only makes SSL methods more efficient but also improves their generalization capabilities, aligning with the broader goals of this thesis in exploring how strategic sparsity can unlock new capabilities in model training and application. The subsequent sections will provide a deeper examination of the methodology, experiments, and results, illustrating how DILEMMA contributes to the evolving landscape of efficient and effective self-supervised learning in computer vision.

3.1 Background

This chapter extends the principles of self-supervised learning (SSL) for image representations discussed in Section 2.1. DILEMMA draws inspiration from classic SSL proxy tasks such as patch location classification [61, 192] and image transformation recognition [95, 132], as outlined in Section 2.1.1. However, it leverages the architecture of Vision Transformers (ViTs) [67] to implement these ideas more effectively, aiming for improved transfer performance.

A key concept not previously discussed is the notion of shape bias versus texture bias in learned representations. Recent work suggests that representations with a shape bias tend to generalize better to downstream tasks [94, 239]. Interestingly, both CNNs and ViTs have been shown to exhibit a texture bias, regardless of whether they are trained in a supervised or unsupervised manner [187]. This insight motivates our approach in DILEMMA, where we aim to enhance shape discriminability in the learned representations.

In the context of ViTs, our method relates to masked token prediction techniques [16, 110, 299] discussed in Section 2.1.3, but focuses on detecting misplaced tokens. This approach shares similarities with methods identifying corrupted tokens in language models [46, 47]. As highlighted in Section 2.1.5, token dropping has emerged as an effective technique for improving ViT training efficiency. DILEMMA extends this concept by introducing randomized token dropping ratios, bridging the gap between pre-training and downstream task distributions.

3.2 Method

Let us define an image sample as $x \in \mathbb{R}^{H \times W \times C}$, *i.e.*, x has $H \times W$ pixels and C color channels. We apply two data augmentations [102] to x and obtain $\hat{x}_1$ and $\hat{x}_2$. Similarly to ViT, each input $\hat{x}_1$ and $\hat{x}_2$ is divided in 14×14 tiles, flattened and projected to N tokens $t_{1,i}, t_{2,i} \in \mathbb{R}^D$, $\forall i \in U \doteq \{1, \ldots, N\}$, through a linear projection. We then combine each token $t_{\cdot,i}$, with a positional embedding $p_i \in \mathbb{R}^D$, which can be either learned or fixed.

As in MoCoV3 [39], we define a *Student* S and a *Teacher* T ViTs [67], where the Teacher, also called *momentum encoder*, is obtained through the exponential moving average (EMA) of the Student's weights (thus, it is not trained). The Teacher receives as input all the tokens $t_{1,1}, \ldots, t_{1,N}$ with the corresponding positional embeddings $p_1, \ldots, p_N$. The Student instead

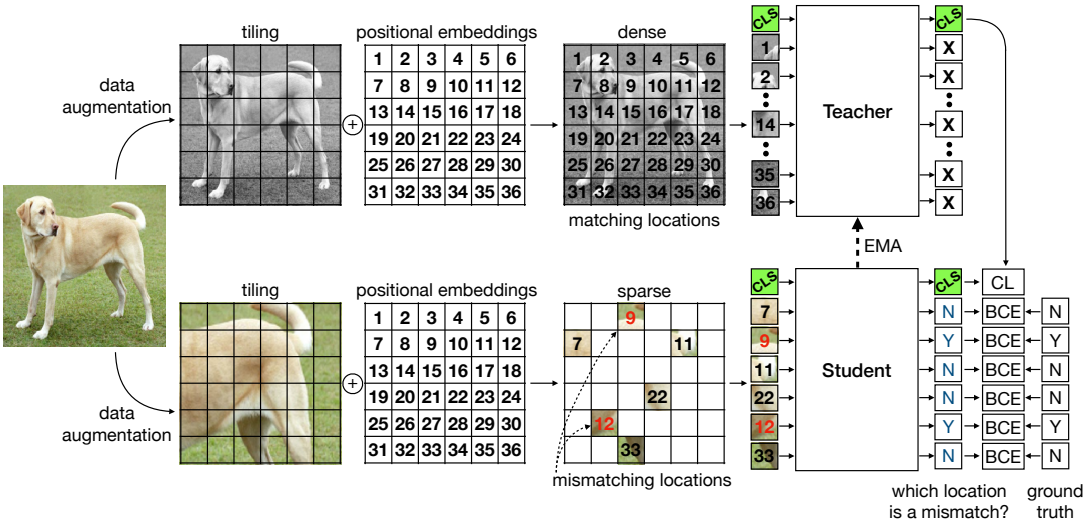


Figure 3.2: **DILEMMA training overview.** A sample image is augmented twice and split into tiles (we use a 14×14 grid). The Teacher network takes the complete set of tiles as input (dense) and without mismatches in the positional embeddings for each token. The Student takes only a subset of the tiles as input (sparse) and some tiles have incorrect positional embeddings. The Student is then trained under two losses: one is the contrastive loss of the class tokens (CLS) between the Teacher and the Student, and the other is the DILEMMA binary cross-entropy for each token.

receives as input a sparse set $M \subset U$ of tokens $t_{2,i}$, $i \in M$. For a randomized fraction of these tokens $B \subset M$ the corresponding positional embdeddings q_i, $i \in M$ are incorrect, *i.e.*, $q_i \doteq p_i$ if $i \in M \setminus B$ and $q_i \doteq p_j$ with $j \in U \setminus M$, if $i \in B$. We call the ratio $\theta = {}^{|B|}/_{|M|} \in [0, 1]$, between the cardinalities of B and M, the *probability of a positional embedding mismatch*. We choose a different M and B sets for each sample at each iteration. We define a set of ground truth labels $y_i = 0$ (N) if $i \in M \setminus B$ and $y_i = 1$ (Y) if $i \in B$. The i-th output token from the Student is denoted with $S_i(\{q_j \oplus t_{2,j}\}_{j \in M})$. We indicate the extra classification token with $i = 0$ both at the input and output. Also, $q_0, p_0 = 0$, *i.e.*, no location encoding.

Now we are ready to introduce the DILEMMA loss (see also the whole training method in Fig. 3.2)

$$\mathcal{L}_{\text{DILEMMA}} = \mathbb{E}_x \Big[\sum_{i \in M} y_i \log \left(\sigma \left(Y S_i \left(\{q_j \oplus t_{2,j}\}_{j \in M \cup \{0\}} \right) \right) \right) + (1 - y_i) \log \left(1 - \sigma \left(Y S_i \left(\{q_j \oplus t_{2,j}\}_{j \in M \cup \{0\}} \right) \right) \right) \Big] \quad (3.1)$$

where $\mathbb{E}[\cdot]$ is the expectation over image samples, σ is the sigmoid function and Y is a linear projection.

Because of the sparsity in the input to the Student network, we also obtain a computational benefit. When we increase the sparsity of the input, we can also increase the mini batch size

to fully utilize the GPU RAM. This is particularly significant with ViTs, because of their quadratic scaling with the number of tokens (the memory usage is $O(N^2)$). The fact that we can significantly increase the mini batch size is particularly effective with contrastive learners. Moreover, in this way it is also faster to train our model, because the average mini batch size is much larger than when using dense inputs (in our case it is $2.5\times$ more).

3.2.1 Combining DILEMMA and Contrastive Learning

The DILEMMA loss can be integrated with other SSL losses. Here we describe the integration with the contrastive loss, but other choices follow an identical procedure.

The contrastive loss is defined as

$$\mathcal{L}_{\text{CNT}} = \mathbb{E}_x\left[L_{\text{CE}}\left(S_0\left(\{q_j \oplus t_{2,j}\}_{j\in M\cup\{0\}}\right), T_0\left(\{p_j \oplus t_{1,j}\}_{j=0,\dots,N}\right)\right)\right], \tag{3.2}$$

where

$$L_{\text{CE}}\left(A, V\right) = -2\tau \sum_n z_n \log \text{softmax}\left(\frac{A_n^\top V}{\tau}\right) \tag{3.3}$$

and A and V are $G \times m$ matrices, with m the minibatch size and G the vector size after the projection Y (see eq. (3.1)), z_j is the one-hot vector with 1 at the j-th position and the index n indicates the class token within the minibatch.

When we combine both the DILEMMA and the contrastive losses into a single cost we obtain

$$\mathcal{L}_{\text{UNION}} = \lambda_{\text{DILEMMA}}\mathcal{L}_{\text{DILEMMA}} + \mathcal{L}_{\text{CNT}}, \tag{3.4}$$

which we minimize and where $\lambda_{\text{DILEMMA}} > 0$ is a hyper parameter which we always set to 0.4 .

3.2.2 Implementation

Architecture. We use Vision Transformers (ViT) [67] with a patch size of 16×16 pixels and an input image size of 224×224 pixels, which gives a total of $(224/16)^2 = 196$ tokens. Due to computational limitations, we mostly use the small variant of the Vision Transformer (ViT-S) which has 12 transformer blocks and 384 channels. For the three baselines: 1) For MoCoV3 [39] experiments, we use 12 attention heads in each attention layer as specified in the official implementation. This is different from most ViT-S implementations, which use 6 heads. This does not change the total number of parameters of the model, but incurs a speed penalty. We use a 3-layer MLP for the projection and prediction heads with synchronized batch normalization. We also freeze the weights of the patch embedding layer for better stability; 2) SimCLR [33] experiments are also conducted with the exact same settings, but without a teacher network and instead both augmentations are sparisified, misplaced and then fed to

the student network; 3) For DINO [29] we used the official implementation and, whenever multi-crop is used, we have disabled random sparsity and used constant sparsity for the large crops and no sparsity for the small crops (96×96 images).

Pre-training Setup. For our main model, we pre-train DILEMMA on ImageNet-1K [56] with the exact same hyper-parameters of MoCoV3 using three GeForce RTX 3090 GPUs for 100 epochs with a base batch size of 345. We set λ_{DILEMMA} to 0.4 and the probability of positional embedding mismatch $\theta = 0.2$. We use sparsity ratios of 0%, 40%, 55%, 65% with $1\times$, $2\times$, $3\times$, $4\times$ base batch size and disable the DILEMMA loss when the input is dense.

To show the compatibility of the proposed method with other SSL methods, we also added two short runs for SimCLR and DINO with multi-cropping. For the DINO experiments we used ViT-Base to show that DILEMMA scales to larger models. Since input sparsity allows for faster training, we also report results of DILEMMA variants with equal training time as the baselines.

Linear Probing. To evaluate the pre-trained features for image classification, we train a simple linear layer on top of frozen features, without any data augmentation (Linear_F). Note that it is different from the standard linear probing, and we opt to use this method for its simplicity and speed. It is also more aligned with the end goal of representation learning. In all the linear probing experiments, we use the embedding of the CLS token of the last layer and perform a coarse grid search over learning rates, batch sizes and whether to normalize the data before feeding it to the linear layer or not (similarly to the added BatchNorm layer [127] in MAE [110]). In contrast, DINO [29], obtains its representation by concatenating the CLS token of the last four attention layers of the network.

3.3 Experiments

We evaluate the use of DILEMMA on several datasets, compare it to state-of-the-art (SotA) SSL baselines, and perform ablations to show the role of each loss component. In each table, where we compare to an SSL baseline, we indicate the baseline with a method name (*e.g.*, MoCoV3 [39]) and use a $+\{$**DILEMMA/sparsity**$\}$ to indicate that the baseline immediately above is combined with just sparsity or with the DILEMMA loss, which includes sparsity. We compare these two cases to show the added benefit of the DILEMMA positional classification loss over the lone sparsity.

3.3.1 Classification on ImageNet-1K

We show that DILEMMA leads to better representations for ImageNet-1K than prior SotA methods. Since this dataset has been used as a reference in SSL, it allows an easy comparison

Table 3.1: **ImageNet classification.** The evaluation uses k-NN and linear probing with a ViT-S/16 or, where indicated, a ViT-Base/16 architecture. The $^{\uparrow}$ models are trained for a number of epochs, such that the total training time (see column Time) is the same as for the baseline methods. BS stands for Batch Size. † models are trained with multi-crop. * indicates ViT-Base/16 models.

Method	Epochs	Time	BS	k-NN	Linear$_F$	Linear
SimCLR	30	15.7h	512	41.46	50.21	-
+Sparsity	30	12.2h	512	41.11	49.73	-
+DILEMMA	30	12.2h	512	**41.90**	**50.71**	-
DINO*†	45	120.9h	192	61.35	65.46	-
+Sparsity*†	45	90.7h	192	62.33	68.49	-
+DILEMMA *†	45	90.7h	192	62.48	68.55	-
+DILEMMA $^{\star\dagger\uparrow}$	60	121.0h	192	**63.74**	**69.43**	-
MoCoV3	100	102.8h	345	59.68	63.62	65.1
+Sparsity	100	68.4h	345	61.64	65.16	-
+DILEMMA	100	68.4h	345	61.97	65.62	66.6
+Sparsity$^{\uparrow}$	150	102.6h	345	63.27	67.07	-
+DILEMMA $^{\uparrow}$	150	102.6h	345	**64.69**	**68.03**	-
MoCoV3	300	-	4096	67.90	72.72	73.2
DINO	300	-	1024	67.9	-	72.5
DINO†	800	-	1024	74.30	75.74	77.0
Supervised	300	-	1024	-	-	79.8

with previous work. In all tested cases, DILEMMA shows a consistent and significant improvement over the baseline it has been integrated with. Notice that the improvement due to the positional loss, relative to the use of sparsity, becomes more significant with a longer training.

k-NN and Linear Probing. In Table 3.1, we evaluate the quality of the ImageNet-1K pretrained features. We either use a weighted k nearest neighbor classifier (we always use $k = 20$) [272] or a simple linear layer on top of a frozen backbone and frozen features. Since the use of sparsity has the added benefit of reducing the computational load at each iteration, we also show the actual training time. For example, with a ViT-Base/16 model and multi-crop, DINO + DILEMMA (denoted with the $^{\uparrow}$ symbol) trains for 60 epochs in about the same time DINO trains for 45 epochs. This gives a significant advantage in performance. Furthermore, DILEMMA outperforms the baseline methods even if trained for the same number of epochs. The improvement under the same number of epochs is about $1 - 2\%$ due to sparsity and

Table 3.2: **Few-shot learning on ImageNet-1K.** Our method significantly outperforms the baseline in both the same training iterations and the same training duration settings. The $^{\uparrow}$ variants are trained for the same duration as the corresponding (non-sparse) baselines. In the Single-Crop case, DINO is shown only as a reference.

		ImageNet-1%		ImageNet-10%	
	Method	k-NN	Linear$_F$	k-NN	Linear$_F$
Single-Crop	DINO	40.60	45.24	52.95	58.35
	MoCoV3	38.48	43.69	50.83	56.08
	+Sparsity	40.02	45.44	52.56	59.06
	+DILEMMA	41.64	47.95	53.15	60.00
	+Sparsity$^{\uparrow}$	42.42	48.34	54.62	61.29
	+DILEMMA $^{\uparrow}$	**45.62**	**51.58**	**56.66**	**62.61**
Multi-Crop	DINO	41.79	46.88	53.00	59.48
	+Sparsity	42.36	48.65	53.61	62.33
	+DILEMMA	42.73	48.81	53.81	62.32
	+DILEMMA $^{\uparrow}$	**43.87**	**50.45**	**55.29**	**63.36**

$0.15 - 0.33\%$ due to the positional classification loss for the k-NN evaluation. Similarly, it is about $2 - 3\%$ due to sparsity and $0.06 - 0.46\%$ due to the positional classification loss for our linear probing. Notice that the boost due to the positional classification loss becomes more significant with more epochs (*e.g*, for MoCoV3 and under the same running time, the k-NN evaluation shows a boost of 3.59% due to sparsity and an additional 1.42% due to the positional classification).

For the sake of completeness, we have also included best reported numbers for ViT-S/16 with significantly larger batch sizes and more training epochs.

Few-shot learning. In Table 3.2, we simulate transfers to small datasets. With reference to ImageNet, we use the model pre-trained on the whole unlabeled dataset, train a linear layer on top of the frozen features of the 1% or 10% subsets [33] and then evaluate the results on the whole validation set. The results show that adding DILEMMA to MoCoV3 or DINO yields a more label-efficient representation than with the corresponding baselines. Notice that in this implementation DILEMMA is based on MoCoV3, which, as was observed in DINO [29], has a consistently worse k-NN accuracy than DINO. Nonetheless, the addition of DILEMMA can more than compensate for the performance gap.

3.3.2 Downstream Tasks

We evaluate DILEMMA on several datasets to assess its generalization capability across different classification and detection tasks. While DILEMMA improves the performance over

Table 3.3: **Transfer learning for image classification.** Our method demonstrates superior performance across various datasets, significantly improving over the baselines. The ↑ variants are trained for the same duration as the corresponding (non-sparse) baselines. **—Position** refers to the MoCoV3 case where the input tokens lack corresponding positional embeddings.

Dataset	*MoCoV3*	*-Position*	*+Sparsity*	*+DILEMMA*	*+Sparsity*↑	*+DILEMMA*↑	*DINO*	*+Sparsity*	*+DILEMMA*	*+DILEMMA*↑
Aircraft	38.70	16.29	43.29	44.43	44.64	**46.02**	45.66	46.83	46.86	**48.60**
Caltech_{101}	87.35	60.79	89.25	89.55	89.89	**90.29**	88.30	89.16	89.58	**89.66**
Cars	28.72	5.88	40.14	42.32	40.21	**43.44**	47.07	48.45	49.04	**50.85**
CIFAR_{10}	91.97	31.36	92.28	93.03	93.53	**94.20**	90.61	92.47	92.46	**93.39**
CIFAR_{100}	75.09	13.96	77.30	77.56	78.88	**80.05**	74.06	77.00	77.22	**78.85**
DTD	64.63	50.59	65.05	64.47	**65.48**	65.37	66.22	**68.40**	67.50	68.30
Flowers_{102}	91.67	60.51	93.25	93.41	94.21	**94.47**	94.54	94.70	95.20	**95.38**
Food_{101}	67.97	31.64	71.39	71.94	72.96	**74.10**	73.00	74.97	74.90	**75.48**
INat_{19}	33.30	18.12	42.38	43.76	42.84	**44.13**	47.36	51.32	52.51	**52.71**
Pets	84.63	53.34	86.35	85.88	**88.93**	88.53	85.83	85.45	85.91	**86.54**
STL_{10}	95.76	70.61	96.09	96.08	**96.90**	96.75	96.62	96.85	97.06	**97.41**
SVHN	64.56	20.02	64.60	65.41	64.52	**66.30**	53.74	66.65	70.01	**71.47**
Yoga_{82}	56.41	15.25	62.29	63.76	63.24	**64.90**	59.90	61.90	62.87	**64.00**
Average	67.75	34.49	71.05	71.66	72.02	**72.97**	70.99	73.40	73.93	**74.82**

the baselines in all the datasets, the most significant improvement seems to occur for more shape-based tasks, such as pose classification. The evaluation on object segmentation, which is a dense downstream task, illustrates the representation captured by the non-CLS tokens.

Transfer Learning. In Table 3.3, we evaluate the transfer capability of our representations for image classification on several datasets. We use: Aircraft [177], Caltech_{101} [81], Cars [145], CIFAR_{10} [146], CIFAR_{100} [146], DTD [44], Flowers_{102} [191], Food_{101} [22], INat_{19} [126], Pets [199], STL_{10} [48], SVHN [189], and Yoga_{82} [250]. We train a linear layer on top of the frozen features to accelerate the process. DILEMMA performs well in transfer learning across all datasets and significantly more on datasets with shape-based tasks, such as Yoga_{82} [250] (for yoga position classification).

We also try to measure approximately how much shape matters in each dataset. We evaluate MoCoV3 with tokens without their position embedding. For simplicity, we use the same pre-trained MoCoV3 used throughout the experiments (although one should ideally use

Table 3.4: **Semantic segmentation on ADE20K.** Due to our per-token loss and corresponding feedback, we achieve better dense representations, resulting in notable improvements over the baselines. The $^{\uparrow}$ variants are trained for the same duration as the corresponding (non-sparse) baselines.

Method	Seg. w/ Lin.			Seg. w/ UPerNet		
	mIoU	mAcc	aAcc	mIoU	mAcc	aAcc
MoCoV3	12.44	15.91	65.95	32.13	43.37	76.79
+Sparsity	15.77	19.87	67.70	33.66	45.27	77.44
+DILEMMA	16.81	21.05	67.84	33.79	45.33	77.68
+Sparsity$^{\uparrow}$	15.93	20.03	67.87	34.03	45.90	77.47
+DILEMMA $^{\uparrow}$	**17.11**	**21.48**	**67.98**	**34.98**	**46.73**	**77.97**
DINO	23.51	30.42	68.73	30.64	43.90	74.52
+Sparsity	26.75	34.20	71.61	33.82	47.01	76.56
+DILEMMA	27.78	35.63	72.63	34.11	47.50	76.73
+DILEMMA $^{\uparrow}$	**28.72**	**36.72**	**72.79**	**34.87**	**47.95**	**77.61**

a MoCoV3 trained without position embeddings). We indicate this case with $-$**Position** in Table 3.3. Without position embedding these features are equivalent to a bag of features. We can see that the improvement due to DILEMMA relative to the baseline MoCoV3 follows the corresponding relative degradation due to the bag of features representation. This suggests that DILEMMA tends to generalize better on datasets with shape-based tasks.

Semantic Segmentation on ADE20K. In Table 3.4, we show the evaluation of DILEMMA on semantic segmentation. This task strongly relates to the shape of objects, thus we expect significant improvement from a boost in shape discriminability. The semantic segmentation capability of self-supervised methods is usually evaluated by fine-tuning the model with an extra decoder. For that we use UPerNet [274] on the ADE20K [297] dataset and train the model for 160K iterations with a batch size of 2 for ViT-Base and 8 for ViT-Small. We also follow the evaluation protocol of iBOT [299] and just train a linear layer (for 160K iterations and a batch size of 16) for semantic segmentation with a frozen backend to directly assess the per-token representation. The results show that DILEMMA is also better than the baseline models for dense classification tasks. It yields remarkable mIoU improvements of 4.6% against MoCoV3 and of 5.2% against DINO in the linear settings and under the same training time.

Unsupervised Object Segmentation. In Table 3.5, we evaluate the single frame object segmentation task. We use the mask generated from the attention of the CLS token (thresholded to keep 0.9 of the mass) as in DINO [29], and report the Jaccard similarity between the ground

Table 3.5: **Unsupervised object segmentation.** We show the mean region similarity $\mathcal{J}_m$ and the mean contour-based accuracy $\mathcal{F}_m$ for DAVIS, and the Jaccard similarity for VOC12. The $^{\uparrow}$ variants are trained for the same duration as the corresponding (non-sparse) baselines.

	DAVIS			VOC12
Method	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$	Jac.$_{sim.}$
MoCoV3	58.28	57.46	59.09	46.50
+Sparsity	58.94	57.05	60.83	45.34
+DILEMMA	**60.00**	**57.99**	**62.02**	48.89
+Sparsity$^{\uparrow}$	58.03	56.74	59.33	45.93
+DILEMMA $^{\uparrow}$	59.84	57.98	61.69	**49.36**
DINO	57.01	55.13	58.90	41.60
+Sparsity	56.83	54.84	58.81	40.03
+DILEMMA	**57.60**	**55.39**	**59.80**	39.71
+DILEMMA $^{\uparrow}$	57.25	55.18	59.31	**44.14**

truth and the mask evaluated on the validation set of PASCAL-VOC12 [76]. For the videos we use the DAVIS-2017 video instance segmentation benchmark [204] and by following the protocol introduced in Space-time by Jabri et al. [129] we segment scenes via the nearest neighbor propagation of the mask. In these evaluations, the role of the positional classification loss seems to be more important than sparsity alone.

Humanoid Vision Engine Benchmark. We also use the newly introduced HVE [93] to evaluate our shape bias in Table 3.6. In HVE Shape dataset, the input images are only the depth map of the foreground object which only contains shape information. We see that DILEMMA outperforms the base model which confirms our hypothesis that DILEMMA can focus on shape. For the HVE Texture, only four grey scaled random crops of the foreground object are concatenated and fed as input, so predicting the right class requires high texture discriminability. Results on HVE Texture show that DILEMMA's better shape understanding did not harm the texture discriminability.

Robustness against Background Change. Following the background challenge evaluation metric [273], we compute the classification accuracy of the model on a subset of ImageNet (IN-9) by changing the background and foreground. As shown in Table 3.7, in O/N.F. (Only/No Foreground), M.S/R/N. (Mixed Same/Random/Next), where the foreground is visible or accurately masked out, we outperform the base model. When the foreground is not visible (O.BB. (Only Background with foreground box Blacked out) and O.BT. (Only Background with foreground replaced with Tiled background)) the model performs correctly and does not just rely on the background for image classification.

Table 3.6: **Humanoid Vision Engine benchmark results.** DILEMMA excels in shape recognition on the HVE [93] Shape dataset and maintains strong texture discriminability on the HVE Texture dataset. The $^{\uparrow}$ models are trained for the same total time as the baseline methods.

Method	Shape Accuracy	Texture Accuracy
MoCoV3	80.78	82.66
+Sparsity	82.55	81.78
+DILEMMA	**83.58**	82.11
+Sparsity$^{\uparrow}$	82.72	82.77
+DILEMMA $^{\uparrow}$	83.52	**83.82**
DINO	80.84	79.47
+Sparsity	83.18	81.01
+DILEMMA	83.58	80.79
+DILEMMA $^{\uparrow}$	**83.64**	**81.45**

Table 3.7: **Robustness against background changes.** We evaluate robustness using the background challenge metric [273]. Our models outperform the baselines in scenarios with visible or accurately masked foregrounds, and perform correctly without relying on the background when the foreground is not visible. The $^{\uparrow}$ models are trained for the same total time as the baseline methods.

Method	Background Change							Clean
	M.N.(↑)	*M.R.*(↑)	*M.S.*(↑)	*N.F.*(↑)	*O.BB.*(↓)	*O.BT.*(↓)	*O.F.*(↑)	IN-9(↑)
MoCoV3	64.52	65.68	78.57	38.69	9.41	10.67	77.80	91.65
+Sparsity	65.53	67.75	80.25	38.72	9.48	10.40	78.15	92.52
+DILEMMA	65.19	68.37	79.63	39.19	**8.42**	**9.68**	78.37	92.00
+Sparsity$^{\uparrow}$	66.25	69.60	81.26	40.25	10.99	10.25	80.10	92.77
+DILEMMA $^{\uparrow}$	**68.86**	**71.16**	**81.85**	**40.40**	8.69	10.64	**82.42**	**93.46**
DINO	65.56	68.94	79.95	33.28	9.70	**9.90**	80.99	92.17
+Sparsity	67.68	71.28	82.10	35.23	8.89	11.11	83.26	93.43
+DILEMMA	**69.58**	71.85	82.89	36.02	9.31	10.47	83.75	93.06
+DILEMMA $^{\uparrow}$	69.38	**73.75**	**82.94**	**38.54**	**8.81**	**9.90**	**84.69**	**93.93**

3.3.3 Ablations

In these experiments, we want to validate empirically a number of choices: 1) we ask how much the trained model is robust to occlusions (sparsity) and positional errors; 2) whether the selection of tokens should be random or guided; 3) whether the ratio of dropped tokens should remain constant in time or instead vary; 4) what the relevance of the positional classification

Table 3.8: **Dropping ratio.** A random dropping ratio is better than a constant one.

	IN100		IN-1K	
Sparsity	k-NN	Linear	k-NN	Linear
0% (Dense)	**76.16**	77.50	53.27	58.20
75%	73.98	77.78	52.99	57.90
Random	74.46	**78.82**	**55.71**	**59.55**

Table 3.9: **Mismatch probability.** Too much mismatch between the tokens and their positions hurts performance.

	IN-1K		Yoga82	
θ	k-NN	Linear	k-NN	Linear
0.3	55.34	59.79	34.95	56.63
0.2	**55.63**	**59.84**	**35.94**	**57.26**

Table 3.10: **Mismatch detection (MD).** Detecting misplaced tokens for dense inputs is easily solved, but still improves the model's performance on shape based tasks. Note that adding both MD and Sparsity to the base model is the same as DILEMMA.

	IN-1K		Yoga82		MD
	kNN	Lin.	kNN	Lin.	Acc.
MoCoV3	53.27	58.20	31.60	51.27	-
+MD	54.18	58.78	35.78	54.53	100
+Sparsity	**55.71**	59.55	32.73	50.90	-
+Both	55.63	**59.84**	**35.94**	**57.26**	96.2

Table 3.11: **Variants of the loss.** We evaluate different loss variants and their effectiveness in improving performance over the baseline. While all variants show improvement, DILEMMA proves to be the most effective.

	IN-1K		Yoga82	
Task	kNN	Lin.	kNN	Lin.
None (MoCoV3)	53.27	58.20	31.60	51.27
Pos. Correction	54.77	58.95	35.74	56.15
Partial Jigsaw	**55.72**	59.19	34.77	56.79
Flip Detection	55.69	59.59	35.09	55.00
DILEMMA	55.63	**59.84**	**35.94**	**57.26**

loss is; 5) the impact of the number of positional errors used during training; 6) whether other design variations are more effective than DILEMMA.

Ablation studies are conducted either on ImageNet100 (IN100) or ImageNet-1K (IN-1K). For the smaller dataset we train the dense models for 300 epochs and the sparse models for 450 epochs (with the same hardware and time settings). For IN-1K experiments we train all models for 50 epochs with MoCoV3 unless stated otherwise.

Randomized Dropping Ratio. In Table 3.8, we verify that a randomized dropping ratio is better than a constant one. We conducted two experiments: one on IN100 and one on IN-1K. The results show that a randomized dropping ratio performs better than a constant one. On the more difficult IN-1K dataset, just applying sparsity is worse than using the dense model. Only with a random dropping ratio can the sparse model outperform the dense model.

Mismatch Probability. The probability of a positional embedding mismatch θ is one of the hyper-parameters of DILEMMA. Early in our experiments, we found out that 20% is much better than 15% (which is used by Electra [47]), probably due to the higher information redundancy in images compared to text. In Table 3.9, we show that $\theta = 30\%$ yields worse

Table 3.12: **Training timing and memory usage.** measured by training ViT-Small models with four RTX Geforce 3090 GPUs. MC stands for Multi-Crop. † models use ViT-Base

Method	BatchSize	EpochTime	MaxMem(GB)
SimCLR	640	21:35	23.65
+DILEMMA	1680	18:20 (×0.85)	23.17
MoCoV3	656	49:08	23.41
+DILEMMA	1664	32:11 (×0.65)	23.55
DINO	576	37:57	22.73
+DILEMMA	1184	24:13 (×0.64)	22.79
DINO(MC)†	144	3:07:21	22.85
+DILEMMA †	216	2:15:52 (×0.72)	23.69

performance than the default $\theta = 20\%$.

Position Classification Loss. In Table 3.10, we verify that the position classification loss helps, by training a dense model with position mismatch detection. Surprisingly, even though the Mismatch Detection (MD) (*i.e.*, the average classification accuracy of the token locations – see "MD Acc." in Table 3.10) is easily solved (it achieves 100% in the dense case), the dense model can still improve the performance of the model on a downstream task. The performance improvement for a task like in Yoga_{82}, which requires a better understanding of shape, is quite significant both with the dense and randomized sparsity inputs.

DILEMMA Variants. We also tried some variants of DILEMMA. Instead of just detecting the misplaced tokens, we predict the right position (as a classification task of 196 classes). The other variant, which we call *Partial Jigsaw*, is to feed some tokens without position encoding and ask the network to predict their position given the other (sparse) correctly position-encoded tokens. Lastly, instead of corrupting the position, one can corrupt the content of a patch. Instead of using complex methods like inpainting we simply horizontally flip some of the patches and use the binary cross-entropy as our loss. Table 3.11 shows that even though all of these methods do help in terms of shape discrimination, DILEMMA is the one with the best performance both on IN-1K and Yoga_{82}.

Timing. To show the efficiency of the proposed method, we ran SimCLR, MoCoV3, DINO with and without multi-crop on 4 GPUs and reported the epoch times in table 3.12.

Token Dropping Policy. In Table 3.13, we compare the case of dropping the tokens that are less important based on the attention of the teacher network [161] compared to randomly dropping the tokens. Results show that simple random dropping works well and there is no

Table 3.13: **Token dropping policy.** Simply dropping the tokens randomly is better than using the importance of tokens (based on CLS attention of the teacher).

Policy	k-NN	Linear
Importance	71.88	76.76
Random	**73.98**	**77.78**

Table 3.14: **Combining DILEMMA with MAE.** Feeding wrongly positioned tokens to the encoder of MAE and detecting them, improves the representation.

Method	Linear	Finetune
MAE	37.30	82.60
+DILEMMA	**39.06**	**83.30**

Table 3.15: **Longer pretraining on IN-100.** Training for longer does not close the gap between baseline and DILEMMA.

Method	300 Epochs	1000 Epochs
MoCoV3	77.50	79.76
+DILEMMA	**78.82**	**81.26**

need to introduce extra complexity to the policy.

Combining with MAE. To show the general applicability of our proposed method to masked models, we misplaced some of the MAE [110] inputs and added DILEMMA loss to the *encoder* of MAE in addition to the reconstruction loss of the decoder. Both MAE and DILEMMA are trained for 200 epochs on IN-100 (using the exact same hyperparameters of the official repository) and results in table 3.14 show that we can outperform MAE both in terms of linear probe and finetuning.

Longer Pretraining. We pretrain MoCoV3 and DILEMMA for 1000 epochs on IN-100 and evaluate their linear performance to see whether the benefits of DILEMMA still hold for longer pretrainings. Results in table 3.15 show that indeed DILEMMA always performs better than the baseline even with longer pretraining.

Weaker Data Augmentations. One of the most important factors for the performance of contrastive learners is the data augmentation. In this short experiment (50 epochs of pretraining, and 70 epochs of linear training) we only used random resized cropping (like MAE [110]) on IN-1K for both MoCoV3 and DILEMMA. Linear probe accuracy of DILEMMA is **44.48%** and for MoCoV3 it is **29.65%** (Note that a 100 epoch pretrained ResNet-50 [108] with SimCLR [33] gets 33.1% accuracy). This huge gap shows that DILEMMA is a generic method for representation learning and does not completely depend on the contrastive component of the loss.

3.4 Discussion

In this chapter, we introduced DILEMMA, a novel self-supervised learning method that addresses the challenges of computational efficiency and representation quality in vision transformers through strategic use of sparsity. By integrating a shape-sensitive regularization

loss with dynamic input sparsification, DILEMMA demonstrates how thoughtful application of sparsity can enhance both the efficiency and effectiveness of SSL algorithms.

Our approach significantly advances the state-of-the-art in SSL by improving shape discriminability in learned representations while simultaneously reducing computational demands. The binary classification loss for detecting correct/incorrect positions of object parts, combined with randomized input sparsity, proves to be a powerful technique for fostering shape discrimination. This aligns with our thesis's broader goal of leveraging sparsity to unlock new capabilities in model training and application.

The teacher-student architecture employed in DILEMMA not only reduces storage and computing resources but also provides a more robust reference for the student network across different sparsity patterns. This innovative approach bridges the gap between training and test data distributions, addressing a key challenge in SSL identified in our introduction.

DILEMMA's performance improvements across established SSL methods such as MoCoV3 [39], SimCLR [33], DINO [29], and MAE [110] under equivalent computational budgets underscore its versatility and potential for broad adoption. These results validate our hypothesis that incorporating shape-sensitive regularization into state-of-the-art SSL methods can lead to enhanced representation learning, particularly beneficial for shape-based tasks like pose classification.

As we transition to the next chapter on SCALE, we extend the principles of sparsity explored in DILEMMA to the more complex domain of video analysis. This progression allows us to address the challenges of computational efficiency, scalability, and representation quality in the context of high-dimensional temporal data, further advancing our exploration of efficient self-supervised visual representation learning via sparsity.

Chapter 4

SCALE: Spatio-Temporal Crop Aggregation for Video Representation Learning

Material in this chapter is based on: Sameni, Sepehr, Simon Jenni and Paolo Favaro. "Spatio-Temporal Crop Aggregation for Video Representation Learning." *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France, 2023, pp. 5641-5651. https://doi.org/10.1109/ICCV51070.2023.00521

Building upon our exploration of sparsity in self-supervised learning (SSL) from the previous chapter, we now turn our attention to the challenging domain of video analysis. This transition amplifies the core challenges of SSL identified in our introduction: computational efficiency, scalability, and the quality of learned representations. In the context of video data, these challenges are particularly acute due to the high-dimensional nature of video content and the complexities of temporal dependencies.

The fundamental question we address is: How can we leverage sparsity and efficient processing techniques to overcome the computational demands and scalability issues inherent in video SSL, while simultaneously improving the quality and generalizability of learned representations? This chapter introduces SCALE (**S**patio-temporal **C**rop **A**ggregation for video representation **LE**arning), a novel approach that directly tackles these challenges while demonstrating how strategic sparsity and efficient processing can enhance model capabilities, particularly in long-term video understanding.

Recent advancements in SSL have shown that pre-training with unlabeled data can surpass supervised pre-training on various downstream tasks [9, 243]. However, SSL methods for video representation learning present fundamental scalability challenges [123, 214, 244, 268]. Our work builds upon the insight that breaking down video processing into multiple steps,

utilizing pre-trained general-purpose models, can lead to more efficient and scalable video analysis [29, 83, 102, 109, 173, 212, 248].

SCALE addresses two key questions: 1) Can we further improve the performance of pre-computed video features by training a model on top of them? 2) Can such training be made computationally scalable? Our approach answers both questions affirmatively by leveraging four key strategies: input sparsity, output sparsity, dimensionality reduction, and the use of a pre-trained backbone.

Input sparsity is achieved by extracting a sparse set of clips from a video, reducing computational load and memory requirements [3, 7, 97, 110, 244]. This approach extends the concept of token sparsity introduced in the previous chapter to the spatio-temporal domain. Output sparsity further reduces computational cost by using a sparse reconstruction output [16, 237, 257, 299]. We work in the latent space to reduce the dimensionality of both input and output data [62, 248, 299]. Finally, we exploit SSL pre-trained models as backbones to reduce training time and speed up processing per iteration.

SCALE integrates these components through two novel pseudo-tasks: Masked Clip Modeling (MCM) and a global feature token task. MCM reconstructs video clip embeddings as in masked autoencoders [244], while the global feature task trains the model to output a summary feature for a set of clips via contrastive learning. Both tasks utilize contrastive losses to enhance the discriminability of individual clip embeddings and obtain a global video representation.

Our method consistently improves pretrained features, with more evident gains considering the computational cost of other methods. For instance, SCALE outperforms VideoMAE [244] and ρBYOL [83] on Kinetics400 with significantly reduced computational time.

Our contributions can be summarized as follows:

- We propose SCALE, a novel and highly scalable video representation method that is trained via novel pseudo-tasks on sets of video clips, in contrast to existing methods that work only with pairs of clips at a time [83, 212, 214].
- We demonstrate significant performance improvements in k-NN (retrieval), linear, and nonlinear probing across a wide range of datasets for action classification (UCF [233], HMDB [147], SSv2 [101], K400 [139]) and long-form video understanding (LVU [269]).
- We achieve consistent transfer learning performance improvement across diverse state-of-the-art pre-trained backbones (architectures, scale, and pre-training tasks). Notably, our nonlinear probed model even outperforms fully fine-tuned SVT [212] on HMDB [147].

By focusing on efficiency and scalability, SCALE not only advances our understanding of SSL for video representation but also aligns with the broader goals of this thesis in exploring how strategic sparsity can unlock new capabilities in model training and application. The subsequent sections will dive deeper into the methodology, experiments, and results, illustrating how SCALE contributes to the evolving landscape of efficient and effective self-supervised learning in video analysis.

4.1 Background

This chapter builds upon the foundations of self-supervised learning (SSL) for video representations, as detailed in Section 2.2. Our approach, SCALE, focuses particularly on long-term video understanding, addressing limitations in existing methods that often struggle to capture long-range temporal dependencies, as discussed in Section 2.2.4.

A key concept not previously elaborated is the use of continuous representations for encoding spatial and temporal positions in videos. Inspired by recent advancements in 3D scene representation, particularly Neural Radiance Fields (NeRF) [180], SCALE adopts a Multi-Layer Perceptron (MLP) to encode clip positions. This approach contrasts with the fixed grids commonly used in existing methods, allowing for more flexible and efficient processing of long video sequences.

As highlighted in Section 2.2.5, input sparsity techniques have emerged as effective strategies for improving the efficiency of video SSL models. SCALE extends these concepts by combining sparse sampling with our continuous representation approach. This enables efficient processing and aggregation of information from arbitrarily sampled space-time crops across long videos, addressing the critical challenge of bridging the gap between short-term feature learning and long-term video understanding in the field of video SSL.

4.2 Method

To describe SCALE, we first define some basic notation and functions, including a general contrastive loss notation that we use for all training losses.

4.2.1 Notation

We use lower-case letters (e.g., z) for generic vectors and capital letters (e.g., Z) for their sets. The expression $a \copyright b$ denotes the concatenation of a and b. We avoid indicating the parameters of networks (usually denoted by θ) if their presence and role are clear from the context. During the training of neural networks, at each iteration, we sample a minibatch of videos. All equations in the next sections are written for a single video in the minibatch. Although not explicitly indicated, all contrastive losses use the other videos in the minibatch as negatives.

4.2.2 Contrastive Loss

InfoNCE is a powerful method for representation learning [248] that can be used to maximize the mutual information between two variables. Because we use this loss between different variables throughout our method, we introduce here a unified notation. Let the paired sets A and B have N elements each, $A = \{a^i\}_{i=1}^N$ and $B = \{b^i\}_{i=1}^N$, where a^i are vectors of dimension

d_A and b^i are vectors of dimension d_B. We also introduce two Multi Layer Perceptrons (MLP), parameterized with θ_A and θ_B, to project these vectors onto a common space of dimension d. After feeding the elements a^i and b^i to the MLPs and normalizing them, we obtain

$$\tilde{a}^i = \frac{\text{MLP}_{\theta_A}(a^i)}{\|\text{MLP}_{\theta_A}(a^i)\|} \quad \text{and} \quad \tilde{b}^i = \frac{\text{MLP}_{\theta_B}(b^i)}{\|\text{MLP}_{\theta_B}(b^i)\|}, \tag{4.1}$$

where $\|\cdot\|$ denotes the L_2 norm. We define the per-element loss based on the relative similarity of $\tilde{a}^i$ and $\tilde{b}^i$, and by using a temperature τ

$$\tilde{\ell}^i(A, B, \theta_A, \theta_B) = -\log \frac{\exp\left(\frac{\tilde{a}^i \cdot \tilde{b}^i}{\tau}\right)}{\sum_{j=1}^{N} \exp\left(\frac{\tilde{a}^i \cdot \tilde{b}^j}{\tau}\right)}. \tag{4.2}$$

We then make the loss symmetric [209] by using

$$\ell^i(A, B, \theta_A, \theta_B) = \tilde{\ell}^i(A, B) + \tilde{\ell}^i(B, A). \tag{4.3}$$

Finally, we define the contrastive loss $\mathcal{L}_{\text{cntr}}(A, B, \theta_A, \theta_B)$ as the mean of ℓ^i

$$\mathcal{L}_{\text{cntr}}(A, B, \theta_A, \theta_B) = \frac{1}{N} \sum_{i=1}^{N} \ell^i(A, B, \theta_A, \theta_B). \tag{4.4}$$

As mentioned earlier on, for simplicity, in the rest of the paper we will not indicate the parameters of the MLPs, and simply write $\mathcal{L}_{\text{cntr}}(A, B)$ or $\ell^i(A, B)$.

4.2.3 Training SCALE

In our method, we integrate 4 principles to drastically reduce the computational complexity of learning a video representation: Input sparsity, output sparsity, dimensionality reduction, and use of a pre-trained backbone. Moreover, we introduce two pseudo-tasks to train the model. One task is based on the (contrastive) reconstruction of a masked video clip given some context video clips. The second task is to build a global representation that is (contrastively) invariant to the set of input sampled video clips. The overall training scheme of SCALE is illustrated in Figure 4.1.

Input sparsity. As a first step, rather than processing a whole video, we collect a sparse set of short video clips from the same video. Given a video $V \in \mathbb{R}^{H \times W \times T \times 3}$, where H, W and T are the height, width, and duration (in frames) of the video, we sample $2 \times K$ clips. We divide the clips into two sets randomly. Each clip in the first set $V_i^1 \in \mathbb{R}^{H_i^1 \times W_i^1 \times T_i^1 \times 3}$, with $i = 1, \ldots, K$, is obtained at the spatio-temporal location X_i^1, Y_i^1, Q_i^1 with different data augmentations and dimensions H_i^1, W_i^1 and T_i^1. Similarly, we denote the second set of clips V_i^2, for $i = 1, \ldots, K$. We also normalize their coordinates relative to the video dimensions

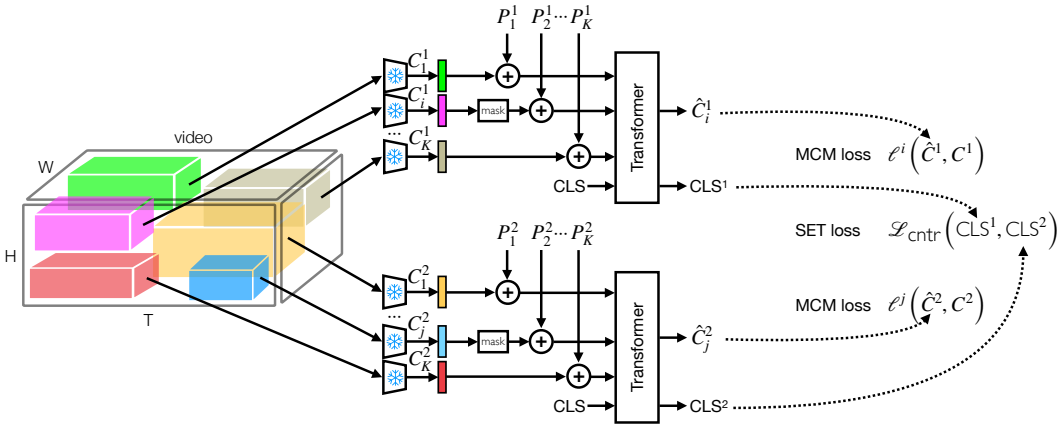


Figure 4.1: **Video Representation Learning with SCALE.** For each video, SCALE extracts two sets of video clips $V_1^1, \ldots, V_K^1$ and $V_1^2, \ldots, V_K^2$. Each video clip is processed separately through a frozen backbone $\mathrm{E}_{\text{❄}}$ and results in encoded video clips $C_1^1, \ldots, C_K^1$ and $C_1^2, \ldots, C_K^2$. Then, a random set of encodings in each set is masked and reconstructed at the output of the predictor network (a transformer) (ℓ^i). The predictor network is also fed a class token CLS. The corresponding output token encodes a summary CLS^m of the m-th set of video clips. The objective for these summary tokens is to be similar only when encoding video clips from the same video ($\mathcal{L}_{\text{SET}}$).

and embed them onto a feature vector P_i^j by feeding them to a learnable MLP. We denote these embeddings

$$P_i^j = \text{MLP}\left(\left[\frac{X_i^j}{H}, \frac{Y_i^j}{W}, \frac{Q_i^j}{T}, \frac{X_i^j + H_i^j}{H}, \frac{Y_i^j + W_i^j}{W}, \frac{Q_i^j + T_i^j}{T}\right]^\top\right), \tag{4.5}$$

where $j = 1, 2$ and $i = 1, \ldots, K$.

Dimensionality reduction and pre-trained backbone. To reduce the dimensionality of each video clip, we feed them independently to a frozen encoder $\mathrm{E}_{\text{❄}}$, to obtain the encodings $C_i^j = \mathrm{E}_{\text{❄}}(V_i^j)$, where $j = 1, 2$ and $i = 1, \ldots, K$. Our framework is encoder-agnostic and thus can work with encoders obtained through different training schemes (supervised, contrastive, or autoencoder). In addition to reducing the dimensionality, we make the training even more scalable by using pre-trained and frozen encoders. Note, however, that if performance is the main goal, it is possible to also train a sparse backbone end to end with multiple clips. Thanks to *token dropping*, one can drop up to 95% of the tokens [97] and still build a good representation.

Output sparsity. As a self-supervised signal for our video representation learning, we use a (contrastive) reconstruction loss. To reduce the computational cost, instead of predicting the features for the whole video [84, 97, 244, 257] (asymmetric decoding), we only reconstruct

a *sparse* set of masked clips. Our reconstruction objective is based on the observation that video signals carry a lot of redundancy. Hence, we introduce a model, the *predictor network*, to predict masked video clip embeddings given the other video clip embeddings (the context). We follow the general approach of BERT [58] but implement the predictor network as a masked autoencoder, where the reconstruction is based on a contrastive loss. The loss is applied only to a sparse set $M^1 \subset \{1, \ldots, K\}$ of masked video clips. These clips are replaced by a learned MSK token. All embeddings C_i^1, including the masked ones, are added to their corresponding position encoding P_i^1 and are then fed to the predictor network. We also include an additional learnable CLS token as input for the predictor network, which will be used for tasks with multiple video clips. We denote the outputs of the predictor network as $\hat{C}_i^1$ for the tokens corresponding to C_i^1, and as CLS^1 for the token corresponding to CLS. Similarly, we feed as inputs separately from the previous set all the video clips C_i^2 with their corresponding positional embeddings P_i^2, and the same CLS token, and obtain $\hat{C}_i^2$ and CLS^2 respectively (see Figure 4.1 for a visual depiction of these processing steps).

Contrastive reconstruction. Modeling all the details of a masked clip, even in the latent space and even given the redundancy in videos, is a demanding task. Rather than increasing the capacity of our model, since we are aiming for scalability, we keep our predictor network a (relatively) shallow network and use contrastive learning [248]. With contrastive learning, the predicted representation of the masked tokens should only be closer to the original unmasked clip representation (after an MLP projection) than from all other clips from the same video and the rest of the minibatch. Note that the rest of the clips in the same video act as hard negatives in contrastive learning [215]. Also, since we are using a frozen backbone, we can afford to use large minibatch sizes, which is known to be beneficial for contrastive learning [33]. We call this contrastive reconstruction loss the Masked Clip Modeling (MCM) loss

$$\mathcal{L}_{\text{MCM}} = \sum_{i \in M^1} \ell^i(\hat{C}^1, C^1) + \sum_{j \in M^2} \ell^j(\hat{C}^2, C^2). \tag{4.6}$$

Multiple video clips loss. The predictor network outputs features for each video clip that are highly discriminative. This task is similar to that of a masked autoencoder [110] and gives you an enhanced per-clip representation. For many video tasks, we need a global representation for the whole video; for that, we introduce an additional pseudo-task that captures a more global representation of a set of video clips. Our task takes inspiration from contrastive learning methods used in SSL, which yield representations that perform well with linear probing [33]. The loss aims to make the CLS^1 and CLS^2 tokens returned from the predictor network more similar (recall that these two tokens are obtained from two separate groups of video clips extracted from the same video) than to other class tokens from other videos within the minibatch

$$\mathcal{L}_{\text{SET}} = \mathcal{L}_{\text{cntr}}\left(\text{CLS}^1, \text{CLS}^2\right). \tag{4.7}$$

In addition to our contrastive loss (InfoNCE), one can use clustering [29] or regression [102] losses. We choose InfoNCE for simplicity and for better compatibility between the losses. As

Table 4.1: **Training throughput and memory usage.** GPU VRAM usage (GB) and max training speed (samples/s) using one 3090 GPU for ViT_B with varying batch sizes and SSL tasks for videos. $\text{MoCo}^{\text{V3}}_{\text{Sparse}}$ is akin to MSN (Masked Siamese Networks) [7]. OOM indicates Out Of Memory error.

Batch Size	8	19	57	2048	Samples/s
MoCo^{V3}	23.47	OOM	OOM	OOM	8.66
VMAE	11.42	22.80	OOM	OOM	33.54
$\text{MoCo}^{\text{V3}}_{\text{Sparse}}$	5.49	9.27	23.62	OOM	28.83
SCALE	1.21	1.23	1.58	19.52	9224.65

the overall loss, we use the sum of both loss terms (without any weights)

$$\mathcal{L} = \mathcal{L}_{\text{MCM}} + \mathcal{L}_{\text{SET}}, \tag{4.8}$$

4.3 Experiments

We evaluate SCALE on several commonly used action classification datasets for video representation learning. As our performance metric, we primarily use linear probing and nonlinear probing [113]. For the smaller datasets, we also use k-NN classifiers (which are training-free) and demonstrate that the proposed method improves upon both unsupervised and supervised backbones.

4.3.1 Experimental Setup and Protocols

Computational Efficiency: In Table 4.1, we show estimates of the maximum batch sizes and the training throughput for different methods trained with the same computational and memory resources. As can be seen, SCALE is orders of magnitude more efficient than other SotA methods. Also during multi-crop inference, our method only results in an FLOP increase of approximately 0.001%.

Datasets: Following prior work [83, 212, 214] we use Kinetics-400 [139], UCF-101 [233] (split 1), HMDB-51 [147] (split 1), and Something-Something v2 (SSv2) [101] to train and evaluate our models. We also use the LVU benchmark [269] to showcase our long-form video understanding capabilities. Note that almost 35% of LVU videos are not available to download from YouTube anymore; thus, our results are not directly comparable with prior methods.

Pretrained backbones: We use the pretrained checkpoints of ρBYOL [83], SVT [212], and three variants of VideoMAE [244] (base(B), large(L), and fine-tuned base(FT)). We choose ρBYOL for their excellent linear performance, SVT for the usage of ViT [67], and VMAE for showing 1) the applicability of our proposed method to MAE models, 2) the scalability

Table 4.2: **Long-Form Video Understanding Results.** Linear and nonlinear probing accuracies on LVU [269] classification tasks. SCALE shows significant performance improvement compared to baselines, indicating its ability to capture long-form video features. The first two rows are greyed out as direct comparison is not possible due to the unavailability of the full dataset for download.

		Relation	Speak	Scene	Director	Genre	Writer	Year
	SlowFast+NL [258]	52.40	35.80	54.70	44.90	53.00	36.30	52.50
	ViS4mer [128]	57.14	40.79	67.44	62.61	54.71	48.80	44.75
SVT	Linear	64.70	35.77	62.33	37.27	54.35	52.12	28.46
SVT	SCALE $_{\text{linear}}$	73.52$^{+8.82}$	40.65$^{+4.88}$	68.83$^{+6.50}$	46.36$^{+9.09}$	57.94$^{+3.59}$	56.38$^{+4.26}$	**39.23**$^{+10.77}$
SVT	MLP	67.64	39.02	66.23	45.45	56.92	57.44	36.15
SVT	Transformer	70.58	40.65	68.83	47.27	57.17	58.51	36.92
SVT	SCALE $_{\text{ft}}$	**76.47**$^{+5.89}$	**42.27**$^{+1.62}$	**74.02**$^{+5.19}$	**49.09**$^{+1.82}$	**58.97**$^{+1.80}$	**62.76**$^{+4.25}$	**39.23**$^{+2.31}$
ρBYOL	Linear	52.94	38.21	53.24	36.36	51.79	57.44	33.84
ρBYOL	SCALE $_{\text{linear}}$	**67.64**$^{+14.70}$	43.08$^{+4.87}$	66.23$^{+12.99}$	44.54$^{+8.18}$	53.33$^{+1.54}$	60.63$^{+3.19}$	40.00$^{+6.16}$
ρBYOL	MLP	62.35	41.46	62.42	47.27	52.56	59.57	40.00
ρBYOL	Transformer	65.09	44.71	66.23	50.90	53.07	61.70	43.84
ρBYOL	SCALE $_{\text{ft}}$	**67.64**$^{+2.55}$	**45.52**$^{+0.81}$	**71.42**$^{+5.19}$	**51.81**$^{+0.91}$	**55.72**$^{+2.65}$	**65.95**$^{+4.25}$	**46.92**$^{+3.08}$

of our method to larger models, and 3) possibility of using supervisedly fine-tuned models as our backbone. All the models are self-supervisedly pretrained on Kinetics-400, except the fine-tuned VMAE base that was also supervisedly finetuned on Kinetics-400. We also used a backbone pretrained and fine-tuned on SSv2 ($\text{VMAE}^{\text{B}}_{\text{SSv2}}$) for the SSv2 experiment to show the universality of SCALE with respect to the pretraining dataset.

Self-supervised Training: For training data, we extract 16 clips of 16 frames from each video per dataset and save their feature encodings to disk. We use PySlowFast's common data augmentations for that [77]. For evaluation, we follow the 5×3 scheme [82] (uniformly sampling 5 clips from a video along its temporal axis and then taking 3 spatial crops) and extract 15 clips from each video (except for SSv2, where we extract 2×3 clips [255]). As the architecture for the predictor network, we use an encoder-only Transformer [249] and a three-layer MLP (without batch normalization [127]) for the contrastive heads. Unless stated otherwise, we train our models for 500 epochs (for example, training with VMAE^{B} on SSV2 takes 137 minutes with one 3090 GPU) with a batch size of 512 and use all 16 clips. We use Adam [140] with cosine annealing learning rate schedule [174] for optimization.

Evaluation: Since our focus is on efficient and scalable video classification, we always freeze the backbones in our evaluation (as in our self-supervised pretraining) and either train a linear classifier [83, 212] or fine-tune the predictor network (the transformer) with an additional linear head. Therefore, when we refer to fine-tuning (ft), we *only* adapt the nonlinear head (*e.g.*, predictor network) but *not the backbone*. We apply a grid search for the hyper-parameters of the heads covering learning rate, weight decay, batch size, and optimizer type. Similar

Table 4.3: **SSv2 Results.** Linear and nonlinear probing accuracies on SSv2 [101]. We see that both SCALE $_{\text{linear}}$ and SCALE $_{\text{ft}}$ outperform other methods and improve the classification accuracies by a large margin. We also see that SCALE $_{\text{ft}}$, with its better initialization, always outperforms the Transformer. $\text{VMAE}^{\text{B}}_{\text{SSv2}}$ was pretrained and fine-tuned on SSv2.

	SVT	ρBYOL	VMAE^{B}	VMAE^{L}	$\text{VMAE}^{\text{B}}_{\text{ft}}$	$\text{VMAE}^{\text{B}}_{\text{SSv2}}$
Linear	20.30	25.30	18.31	27.94	28.90	70.53
SCALE $_{\text{linear}}$	25.26	27.16	21.24	30.18	33.25	<u>70.63</u>
MLP	21.43	26.46	19.42	27.96	29.83	70.52
Transformer	<u>29.24</u>	<u>30.99</u>	<u>24.26</u>	<u>34.39</u>	<u>35.60</u>	70.57
SCALE $_{\text{ft}}$	**29.68**	**31.83**	**25.25**	**36.34**	**37.38**	**70.69**

to MAE [110], we found that applying a batch normalization layer [127] without affine transformations is beneficial for VideoMAE models. As the linear baseline, we consider the well-established ensembling approach, *i.e.*, we average the softmax predictions of the 15 clips (6 for SSv2) to obtain the final prediction. For models that process multiple clips at once (like ours), we likewise apply a linear softmax head on the concatenation of the individual clip features and the [SET] token before averaging to obtain the final prediction. For the smaller datasets, we also use k-NN classification, where, similar to DINO [29], we always use $k = 20$ and work with l2 normalized representations.

Nonlinear baselines: As SCALE is a nonlinear model, we consider an MLP on top of the frozen backbone as a nonlinear baseline. As a further baseline and to illustrate the effect of our self-supervised pre-training, we consider a Transformer trained on all the clip representations. This Transformer uses the exact same architecture as SCALE, and only differs in the initialization: in the case of SCALE we start from our proposed SSL pre-trained weights instead of random initialization.

4.3.2 Results

LVU: One of the benefits of SCALE is that it can be used to process long videos, even though the backbones were trained on short videos only. To demonstrate the ability to capture long-form video features, we evaluated SCALE on LVU [269], a benchmark that involves seven classification (and two regression) tasks on minute-long videos. Past studies [128, 258] have established that increasing the input's time span enhances accuracy in this challenging dataset. As shown in Table 4.2, our experiments indicate that SCALE can improve the baseline model's performance by a considerable margin. Moreover, we found that fine-tuning SCALE can lead to further enhancements.

SSv2: Multiple classes in SSv2 share similar backgrounds and only differ in motion [122], suggesting that high performance on this dataset demonstrates that the model has captured

Table 4.4: **UCF Results.** Linear and nonlinear probing accuracies on UCF-101 [233]. SCALE $_{\text{ft}}$ outperforms all the other models and, in the case of ρBYOL, even gets performance close to a fully finetuned model. Also, in most cases, SCALE $_{\text{linear}}$ outperforms the fine-tuned Transformer and achieves state-of-the-art results in linear probing (previous SotA using RGB frames was 92.6 [214]). We further see a significant accuracy improvement in k-NN probing, especially for pre-trained MAE-based models.

	SVT	ρBYOL	VMAE$^{\text{B}}$	VMAE$^{\text{L}}$	VMAE$^{\text{B}}_{\text{ft}}$
k-NN	87.20	85.19	35.05	49.14	96.82
SCALE $_{k\text{-NN}}$	89.00	83.47	65.63	76.02	97.38
Linear	91.27	89.55	66.53	84.53	97.91
SCALE $_{\text{linear}}$	<u>92.65</u>	91.43	<u>74.46</u>	86.78	<u>98.14</u>
MLP	91.17	93.60	71.97	<u>87.04</u>	98.04
Transformer	92.20	<u>94.34</u>	68.22	86.30	98.04
SCALE $_{\text{ft}}$	**92.94**	**95.00**	**76.07**	**89.92**	**98.46**
FT$_{\text{reported}}$	93.7	95.4	96.1	-	-

strong motion-related contextual cues [212]. Results in Table 4.3 show that we outperform the state-of-the-art. On this dataset, we see a large performance gap between models that process single clips at a time (Linear and MLP) and the models that work with multiple clips (SCALE and Transformer). We can see SCALE $_{\text{linear}}$ is also outperforming the MLP, showing that SCALE is able to capture motion and long-form temporal features of the video. We even improve the supervised model trained on SSv2 (VMAE$^{\text{B}}_{\text{SSv2}}$).

UCF-101 & HMDB-51: For these smaller datasets, besides linear and nonlinear probing, we also use k-NN probing (see Table 4.4 and Table 4.5). With SCALE $_{k\text{-NN}}$, we see a consistent improvement over the baseline and find that pre-trained MAE-based models greatly benefit from our training. This can be explained by the additional invariance properties introduced through the SET loss term in SCALE training. Across the board, we also see that in the case of linear probing, not only does SCALE $_{\text{linear}}$ outperform Linear, but it also outperforms Transformer, which leverages many more parameters. In the case of SVT, our SCALE $_{\text{linear}}$ also outperforms the best reported linear accuracy on UCF101 (92.7% vs. 92.6% [214]). Finally, SCALE $_{\text{ft}}$ achieves better results than all the nonlinear baselines and even outperforms the fully fine-tuned SVT (68.1% vs. 67.2%).

Kinetics-400: We present our main results on K400 [139] in Table 4.6. Our SCALE $_{\text{linear}}$ with SVT backbone beats the previous state of the art (71.8% vs. 71.5% [83]) and SCALE $_{\text{ft}}$ can even improve the accuracy of VMAE$^{\text{B}}_{\text{ft}}$, which is a strong supervised model, from 81.5% to 81.84%.

Table 4.5: **HMDB Results.** Linear and nonlinear probing accuracies on HMDB-51 [147]. Despite the small size of the dataset, we see that SCALE $_{\text{ft}}$ is outperforming all the other methods, and in the case of SVT, it even outperforms the fully fine-tuned model. We also see that SCALE $_{\text{linear}}$ outperforms the Transformer in most cases with only a single linear layer (the best linear accuracy in the literature is 66.7% [214]). Similar to UCF results, we see a considerable increase in the performance of k-NN classifiers for pre-trained MAE-based models.

	SVT	ρBYOL	VMAE$^{\text{B}}$	VMAE$^{\text{L}}$	VMAE$^{\text{B}}_{\text{ft}}$
k-NN	51.83	49.67	21.96	29.21	72.81
SCALE $_{k\text{-NN}}$	56.01	51.56	37.18	51.30	71.83
Linear	63.07	61.17	45.22	60.26	76.33
SCALE $_{\text{linear}}$	66.33	63.92	**52.15**	62.35	78.36
MLP	63.00	64.77	49.01	62.61	77.45
Transformer	63.98	66.16	47.32	61.50	76.86
SCALE $_{\text{ft}}$	**68.10**	**66.79**	51.89	**64.83**	**79.34**
FT$_{\text{reported}}$	67.2	73.6	73.3	-	-

Following the evaluation setup of self-supervised image representations [29, 33, 299], we also introduce low-shot K400 video classifications by sampling 10 percent of the videos (in a class-balanced way) and training the probes only on those. We still test on the whole evaluation set of K400. This low-shot setting is more aligned with the typical use-case of self-supervised models in which there is abundant unlabeled data for training via self-supervision and a small set of labeled data for fine-tuning. Results in Table 4.7 show that our method is particularly effective in this low-shot setting. While most other nonlinear probes overfit and perform worse than the linear probes, our SCALE $_{\text{ft}}$ does not overfit and clearly outperforms the baselines.

4.3.3 Ablations

In this section, we start from a baseline setup consisting of a two-layer transformer with a hidden size of 256, 20% chance of masking clips, trained with a batch size of 512 for 200 epochs, and using two sets of 8 views for representation learning. Using SCALE $_{\text{ft}}$, we explore different loss functions, masking ratios, number of layers, and finally, the number of views during training and testing. All experiments are performed on UCF and HMDB.

Loss Function: As explained in the method section, we have two loss terms, and each of them can be enabled or disabled for the pretraining. In Table 4.8 we show that having both loss terms is better than the individual loss terms.

Table 4.6: **Kinetics-400 Results.** Linear and nonlinear probing accuracies on Kinetics-400 [139] without any extra data and using RGB frames only. While SCALE $_{\text{linear}}$ is on par with Linear, we observe clear improvements for nonlinear probing in the case of SCALE $_{\text{ft}}$. Note that the best linear accuracy on this dataset (without any extra data) is 71.5 [83] and the best full fine-tuning accuracy is 86.7 [263].

	SVT	ρBYOL	VMAE$^{\text{B}}$	VMAE$^{\text{L}}$	VMAE$^{\text{B}}_{\text{ft}}$
Linear	71.71	68.82	43.50	60.73	81.52
SCALE $_{\text{linear}}$	71.78	68.38	43.96	60.66	81.44
MLP	71.19	69.42	45.48	61.64	81.27
Transformer	72.18	69.28	44.85	62.15	81.70
SCALE $_{\text{ft}}$	**72.38**	**69.63**	**46.15**	**62.67**	**81.84**

Table 4.7: **Kinetics-400 Low-shot Results.** Linear and nonlinear probing accuracies on 10% of Kinetics-400 [139]. SCALE is more robust to the size of the labeled dataset. SCALE $_{\text{ft}}$ does not overfit like the other nonlinear probes (MLP and Transformer) and outperforms the baselines. VMAE$^{\text{B}}_{\text{ft}}$ is greyed out since it was already fine-tuned on the whole labeled dataset and is only reported here for consistency with the other tables.

	SVT	ρBYOL	VMAE$^{\text{B}}$	VMAE$^{\text{L}}$	VMAE$^{\text{B}}_{\text{ft}}$
Linear	66.43	56.43	31.25	48.42	79.79
SCALE $_{\text{linear}}$	65.96	57.74	**34.03**	49.21	79.94
MLP	65.44	58.68	30.47	48.27	79.37
Transformer	64.97	58.95	29.89	48.27	79.47
SCALE $_{\text{ft}}$	**67.01**	**59.52**	33.92	**50.36**	**80.47**

Masking Ratio: Masking ratio is an important hyperparameter and depends on the data modality, for example, BERT [58] uses 15%, MSN [7] uses 30% (for ViT-Base), MAE [110] uses 75%, and VideoMAE [244] uses 90 to 95% masking. Since our clip representations are somewhat abstract representations of the video, we expect the optimal masking ratio to be close to NLP models rather than video MAEs. We have observed a steady decrease in the pretraining task's performance with higher masking ratios, so we only tested low masking ratios in Table 4.9 and found out that 25% is the optimal masking ratio.

Transformer Capacity: We also explore the number of transformer layers and their hidden size in Table 4.10. We can see that having more than one transformer layer is necessary for good results and too few hidden channels can hurt performance. However, there is a trade-off, and deeper transformers can lead to worse performance.

Table 4.8: **Loss Function.** SCALE $_{\text{ft}}$ accuracy with different loss function combinations (the masking ratio here is 20%). We can see that having MCM is always beneficial, and the SET loss is almost always helpful. We use both loss terms for our final model.

Loss Term		UCF-101		HMDB-51	
SET	MCM	SVT	ρBYOL	SVT	ρBYOL
✓	✗	91.80	92.20	64.50	63.59
✗	✓	92.01	**93.81**	62.81	64.05
✓	✓	**93.20**	92.99	**64.57**	**65.61**

Table 4.9: **Masking Ratio.** SCALE $_{\text{ft}}$ accuracy with different masking ratios. We observe best results around 25% similar to NLP models [58] (15%), and different from low-level video models like VideoMAE [244] (90%).

Masking	UCF-101		HMDB-51	
Ratio	SVT	ρBYOL	SVT	ρBYOL
0.15	93.18	93.25	64.37	64.83
0.25	**93.20**	**93.81**	**65.49**	**65.62**
0.35	93.15	93.06	64.18	65.22
0.45	92.96	93.02	63.39	64.96

Number of Views: Finally, we studied the model performance as we changed the number of views and batch size fed to the model. As can be seen in Table 4.11, having more views has a large and consistent impact on the performance, and since we have hard negatives for contrastive loss within the video, we are not too reliant on large batch sizes.

CLIP: We conducted an additional experiment with OpenCLIP-ViT-B/16 [42] on Kinetics400 and SSv2. We utilized only 16 random frames (rather than clips) for pretraining and evaluation. Table 4.12 demonstrates the effectiveness of our method in the weakly supervised setting. However, it should be noted that this approach requires careful exploration. Based on our preliminary results, CLIP gives extremely similar encodings to different frames of the same video, making the MCM task difficult, and the MCM accuracy is considerably lower than the video encoder case.

4.4 Discussion

SCALE represents a significant advancement in self-supervised learning for video representation, extending the sparsity principles introduced with DILEMMA to the more complex domain of video analysis. By leveraging spatio-temporal crop aggregation and novel pseudo-tasks, our approach demonstrates how sparsity can effectively address the challenges of computational efficiency, scalability, and representation quality in high-dimensional temporal data.

Our method's ability to process multiple clips simultaneously while maintaining computational efficiency directly tackles the scalability issues inherent in video SSL. The introduction of Masked Clip Modeling and the global feature token task showcases how sparsity can enhance both local and global video understanding.

Extensive experiments across various action classification and long-form video understanding datasets validate SCALE's effectiveness, consistently outperforming state-of-the-art

Table 4.10: **Transformer Capacity.** SCALE $_{\text{ft}}$ accuracy with different model capacities. Having more than one transformer layer and not too few hidden channels is necessary for the best performance.

Hidden Dim	Num Layers	UCF-101 SVT	UCF-101 ρBYOL	HMDB-51 SVT	HMDB-51 ρBYOL
64	1	-	92.62	-	63.16
128	1	-	92.83	-	63.68
256	1	-	92.86	-	64.39
128	2	92.78	93.52	63.26	65.55
256	2	**93.20**	**93.81**	65.49	**65.62**
512	2	92.57	93.66	**65.68**	64.77
128	3	92.33	93.25	64.83	65.55
256	3	92.75	93.52	65.49	65.16
512	3	92.86	92.93	65.49	64.84

Table 4.11: **Number of Views.** SCALE $_{\text{ft}}$ accuracy with different numbers of clips and batch sizes. More views lead to consistent improvement, and large batch sizes are not necessary because of the hard negatives.

Num Views	Batch Size	UCF-101 SVT	UCF-101 ρBYOL	HMDB-51 SVT	HMDB-51 ρBYOL
4×2	256	92.65	92.83	64.35	64.24
6×2	256	92.70	93.07	64.57	64.37
8×2	256	92.80	93.49	64.64	64.63
4×2	512	92.67	93.49	64.85	64.50
6×2	512	93.18	93.68	64.90	65.35
8×2	512	**93.20**	**93.81**	**65.49**	**65.62**
4×2	1024	92.75	93.36	64.77	65.15
6×2	1024	92.96	93.57	64.96	65.48
8×2	1024	93.07	OOM	65.29	OOM

Table 4.12: **CLIP Results.** Probing accuracies on K400 and SSv2 with OpenCLIP-ViT-B/16 [42] with 16 frames (not clips).

	Linear	SCALE$_{\text{linear}}$	MLP	Transformer	SCALE$_{\text{ft}}$
SSv2	15.22	17.70	16.98	22.92	23.51
K400	68.85	69.59	69.14	70.93	71.81

methods while significantly reducing computational demands. Notably, our approach's ability to improve upon diverse pre-trained backbones underscores its versatility and potential for broad adoption.

SCALE's robust performance in low-shot learning scenarios and its effectiveness in enhancing pre-computed video features address key challenges in video SSL identified in our introduction. The surprising effectiveness of contrastive masked modeling on representations trained to be invariant to spatio-temporal crops opens intriguing questions about the nature of learned representations and the potential role of benign memorization [5].

As we transition to the next chapter on RIVER, we extend the principles of sparsity explored in SCALE to the domain of video generation. This progression allows us to investigate whether these efficient processing techniques can be applied to generative tasks, potentially expanding the scope of sparse representations in video understanding and synthesis. This shift from discriminative to generative tasks represents a significant step in our exploration of efficient self-supervised visual representation learning via sparsity.

Chapter 5

RIVER: Efficient Video Prediction via Sparsely Conditioned Flow Matching

Material in this chapter is based on: Aram Davtyan*, Sepehr Sameni* and Paolo Favaro, "Efficient Video Prediction via Sparsely Conditioned Flow Matching." *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France, 2023, pp. 23206-23217. https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.02126

Building upon our exploration of sparsity in self-supervised learning (SSL) from the previous chapters, we now shift our focus to generative models, specifically in the domain of video prediction. This transition allows us to explore implicit representation learning, where models learn to capture complex world dynamics without explicit supervision. By tackling video prediction, we address a fundamental question: how can we efficiently process and generate high-dimensional temporal data while implicitly learning robust world representations? This chapter introduces RIVER (Random frame conditioned flow Integration for VidEo pRediction), a novel approach that demonstrates how strategic sparsity and efficient processing can enhance model capabilities in video prediction, serving as an implicit world model.

Video prediction, *i.e.*, the task of predicting future frames given past ones, is a fundamental component of an agent that needs to interact with an environment [11]. This capability enables planning and advanced reasoning, especially when other agents are in the scene [86, 87, 276]. More generally, however, a video prediction model that can generalize to new unseen scenarios needs to implicitly understand the scene, *i.e.*, detect and classify objects, learn how each object moves and interacts, estimate the 3D shape and location of the objects, model the laws of physics of the environment, and so on. In essence, these models serve as implicit world models, capturing the dynamics and structure of the environment they observe without explicit supervision.

*Equal contribution

This implicit learning of world representations is particularly valuable because it doesn't require any labeling, making it an excellent candidate for learning from readily available unannotated datasets. The process of predicting future frames naturally leads to rich and powerful representations of videos, as the model must internalize complex spatio-temporal relationships and physical laws to make accurate predictions.

While the literature in video prediction is by now relatively rich [11, 57, 151], the quality of the predicted frames has been achieving realistic levels only recently [107, 119, 252, 282]. This has been mostly due to the exceptional complexity of this task and the difficulty of training models that can generalize well to unseen (but in-domain) data.

One of the key challenges of synthesizing realistic predicted frames is to ensure the temporal consistency of the generated sequence. To this aim, conditioning on as many past frames as possible is a desirable requirement. In fact, with only two past frames it is possible to predict only constant motions at test time, and for general complex motions, such as object interactions (*e.g.*, a ball bouncing off a cube in CLEVRER [283]), many more frames are needed. However, conditioning on many past frames comes either at the sacrifice of the video quality or at a high computational cost, as shown in Figure 5.1. In the literature, we see two main approaches to address these issues: 1) models that take a fixed large temporal window of past frames as input and 2) models that compress all the past into a state, such as recurrent neural networks (RNNs)[10, 11, 57]. Fixed window models require considerable memory and computations both during training and at inference time. Although methods such as Flexible Diffusion[107] can gain considerable performance by choosing carefully non contiguous past frames, their computational cost still remains demanding. RNNs also require considerable memory and computations resources at training time, as they always need to feed a sequence from the beginning to learn how to predict the next frame. Moreover, training these models is typically challenging due to the vanishing gradients [118].

To address these challenges, we propose RIVER, a novel training procedure for video prediction that is computationally efficient and delivers high quality frame prediction. Our approach draws inspiration from recent advancements in diffusion models for image generation [115]. We adapt the idea of sparse conditioning, introduced by 3DiM [262], to video prediction. This allows us to condition the generation of the next frame on a randomly chosen sparse set of past frames during the diffusion process, effectively limiting the computational complexity at both training and test time.

To further enhance efficiency, we compress videos via VQGAN autoencoding [75] and work in the latent space. This design choice has been shown to enable efficient generation of high-resolution images [216]. We also incorporate a refinement network to improve frame quality and correct temporal inconsistencies during post-processing.

A significant performance boost is achieved by adapting Flow Matching [164] to video prediction, resulting in better convergence at training time and improved image quality generation.

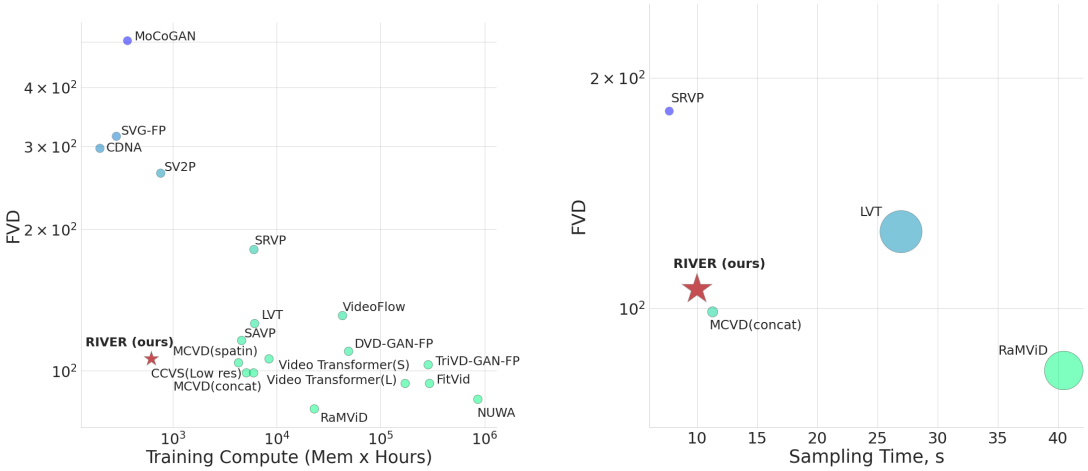


Figure 5.1: **Efficiency and speed comparisons of RIVER.** *Left: RIVER* achieves an ideal trade-off between quality of generated videos (FVD [246]) and compute needed to train the model on the BAIR dataset [72]. This makes research on video models more easily scalable. *Right:* FVD vs. inference speed, the time required to generate a 16 frames long 64×64 resolution video on a single Nvidia GeForce RTX 3090 GPU. The sizes of the markers are proportional to the standard deviation of measured times in 20 independent experiments. We compare to diffusion-based models (RaMViD [119], MCVD [252]), an RNN-based model (SRVP [89]), and a Transformer-based model (LVT [210]). Due to sparse past frame conditioning, RIVER achieves reasonable sampling time. The focus of our method is primarily on efficient training to enable exploring new ideas and architectures, rather than optimizing sampling time, though we still achieve competitive inference speed.

Finally, we introduce a *warm-start sampling* technique to make our method more efficient at inference time, leveraging the fact that in video prediction, content changes slowly over time.

We demonstrate RIVER on common video prediction benchmarks, showing that it performs on par or better than state-of-the-art methods while being much more efficient to train. We also illustrate RIVER's capability in video generation, interpolation, and prediction of non-trivial long-term object interactions, showcasing its potential as an implicit world model.

Our contributions can be summarized as follows:

- We extend flow matching to video prediction;
- We design a model that is efficient to train and use at test time;
- Our approach can be conditioned on arbitrarily many past frames;
- We introduce a warm-start sampling technique for improved efficiency at test time.

By focusing on efficiency and scalability in video prediction, RIVER advances our understanding of world modeling through implicit representation learning. This aligns with the

broader goals of this thesis in exploring how efficient processing can unlock new capabilities in model training and application, particularly in the context of world modeling through video prediction. The subsequent sections will dive deeper into the methodology, experiments, and results, illustrating how RIVER contributes to the evolving landscape of efficient and effective video prediction while implicitly learning powerful world representations.

5.1 Background

This chapter extends the concepts of video prediction discussed in Section 2.2.6, with a focus on recent advancements that form the foundation for RIVER's approach. Our method builds upon three key developments in generative modeling that were not elaborated in the main background chapter.

In the realm of diffusion models, the concept of "implicit" conditioning has emerged as a promising approach. This idea was notably applied by 3DiM [262] to 3D multi-view reconstruction. In this method, instead of conditioning on all views simultaneously, the denoising network is conditioned on a random view at each step of the diffusion process. This approach effectively distributes the conditioning over multiple steps, potentially reducing computational complexity while maintaining model performance. RIVER extends this concept to video prediction, applying it to past frames rather than views.

A significant recent development in generative modeling is conditional flow matching, introduced by Lipman et al. [164]. This approach generalizes diffusion models and has demonstrated faster training convergence and improved results compared to traditional denoising diffusion models. Flow matching provides an explicit mapping from noise instances to image samples, offering a more flexible framework than standard diffusion models. RIVER adopts and adapts this framework for the task of video prediction, leveraging its benefits in the context of generating future video frames.

Researchers have also proposed various methods to accelerate or improve vanilla diffusion models [60, 137, 143]. Of particular relevance to RIVER is the concept of starting the generation process from an intermediate point rather than from pure noise. For instance, CCDF [43] initiates the backward denoising process from an intermediate time step using an initial guess of the final output. Inspired by this idea, RIVER introduces a "warm start" technique within the flow matching formulation. This approach leverages the temporal continuity inherent in video sequences to improve efficiency at inference time.

By integrating these advancements in implicit conditioning, flow matching, and acceleration techniques, RIVER aims to advance the state of the art in efficient and high-quality video prediction, effectively learning and utilizing implicit world representations.

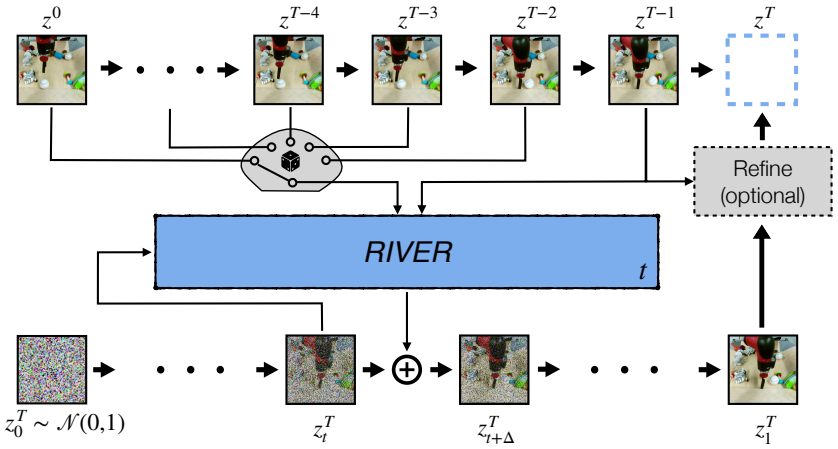


Figure 5.2: **Inference with RIVER.** In order to generate the next frame z^T (top-right), we sample an initial estimate from the standard normal distribution z_t^T (bottom-left) and integrate the ODE (5.2) by querying our model at each step with a random conditioning frame from the past z^c and previous frame z^{T-1} (top). We omitted the encoding/decoding for simplicity.

5.2 Method

Let $\mathbf{x} = \{x^1, \ldots, x^m\}$, where $x^i \in \mathbb{R}^{3\times H\times W}$, be a video consisting of m RGB images. The task of video prediction is to forecast the upcoming n frames of a video given the first k frames, where $m = n + k$. Thus, it requires modelling the following distribution:

$$p(x^{k+1}, \ldots, x^{k+n} \mid x^1, \ldots, x^k) = \prod_{i=1}^{n} p(x^{k+i} \mid x^1, \ldots, x^{k+i-1}). \tag{5.1}$$

The decomposition in eq. (5.1) suggests an autoregressive sampling of the future frames. However, explicitly conditioning the next frame on all the past frames is computationally and memory-wise demanding. In order to overcome this issue, prior work suggests to use a recurrently updated memory variable [31, 194, 251, 260] or to restrict the conditioning window to a fixed number of frames [119, 252, 264, 282]. We instead propose to model each one-step predictive conditional distribution as a denoising probability density path that starts from a standard normal distribution. Moreover, rather than using score-based diffusion models [232] to fit those paths, we choose flow matching[164], a simpler method to train generative models. We further leverage the iterative nature of sampling from the learned flow and use a single random conditioning frame from the past at each iteration. This results in a simple and efficient training. An idea similar to ours was first introduced in [262] for novel view synthesis in 3D applications. In this paper, however, we made some design choices to adapt it to videos.

5.2.1 Latent Image Compression

Although we could operate directly on the pixels of the frames x^i, we introduce a compression step that reduces the dimensionality of the data samples and thus the overall numerical complexity of our approach. Given a dataset of videos D, we train a VQGAN [75] on single frames from that dataset. The VQGAN consists of an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$ and allows to learn a perceptually rich latent codebook through a vector quantization bottleneck and an adversarial reconstruction loss [247]. A trained VQGAN is then used to compress the images to much lower resolution feature maps. That is, $z = \mathcal{E}(x) \in \mathbb{R}^{c \times \frac{H}{f} \times \frac{W}{f}}$, where $x \in \mathbb{R}^{3 \times H \times W}$. Commonly used values for c are 4 or 8 and for f are 8 or 16, which means that a 256×256 image can be downsampled to up to a 16×16 grid. Following [216], we let the decoder $\mathcal{D}$ absorb the quantization layer and work in the pre-quantized latent space. Further in the paper, when referring to video frames we always assume that they are encoded in the latent space of a pretrained VQGAN.

5.2.2 Flow Matching

Flow matching was introduced in [164] as a simpler albeit more general and more efficient alternative to diffusion models [115]. A similar framework incorporating straight flows has also been proposed independently in [4, 168]. We assume that we are given samples from an unknown data distribution $q(z)$. In our case, the data sample z is the encoding of a video frame x via VQGAN. The aim of flow matching is to learn a temporal vector field $v_t(z) : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$, with $t \in [0, 1]$, such that the following ordinary differential equation (ODE)

$$\dot{\phi}_t(z) = v_t(\phi_t(z)) \tag{5.2}$$

$$\phi_0(z) = z \tag{5.3}$$

defines a flow $\phi_t(z) : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ that pushes $p_0(z) = \mathcal{N}(z \,|\, 0, 1)$ towards some distribution $p_1(z) \approx q(z)$ along some probability density path $p_t(z)$. That is, $p_t = [\phi_t]_* p_0$, where $[\cdot]_*$ denotes the push-forward operation. If one were given a predefined probability density path $p_t(z)$ and the corresponding vector field $u_t(z)$, then one could parameterize $v_t(z)$ with a neural network and solve

$$\min_{v_t} \mathbb{E}_{t, p_t(z)} \|v_t(z) - u_t(z)\|^2. \tag{5.4}$$

However, this would be unfeasible in the general case, because typically we do not have access to $u_t(z)$. Lipman *et al.* [164] suggest to instead define a conditional flow $p_t(z \,|\, z_1)$ and the corresponding conditional vector field $u_t(z \,|\, z_1)$ per sample z_1 in the dataset and solve

$$\min_{v_t} \mathbb{E}_{t, p_t(z \,|\, z_1), q(z_1)} \|v_t(z) - u_t(z \,|\, z_1)\|^2. \tag{5.5}$$

Algorithm 1: Video Flow Matching with RIVER

Input: dataset of videos D, number of iterations N
for i in range(1, N) **do**
Sample a video $\mathbf{x}$ from the dataset D
Encode it with a pre-trained VQGAN to obtain z
Choose a random target frame $z^\tau, \tau \in \{3, \dots |\mathbf{x}|\}$
Sample a timestamp $t \sim U[0, 1]$
Sample a noisy observation $z \sim p_t(z \mid z^\tau)$
Calculate $u_t(z \mid z^\tau)$
Sample a condition frame $z^c, c \in \{1, \dots \tau - 2\}$
Update the parameters θ of v_t via gradient descent

$$\nabla_\theta \| v_t(z \mid z^{\tau-1}, z^c, \tau - c\,; \theta) - u_t(z \mid z^\tau) \|^2 \tag{5.7}$$

end for

This formulation enjoys two remarkable properties: 1) all the quantities can be defined explicitly; 2) Lipman *et al*. [164] show that solving eq. (5.5) is guaranteed to converge to the same result as in eq. (5.4). The conditional flow can be explicitly defined such that all intermediate distributions are Gaussian. Moreover, Lipman *et al*. [164] show that a linear transformation of the Gaussians' parameters yields the best results in terms of convergence and sample quality. They define $p_t(z \mid z_1) = \mathcal{N}(z \mid \mu_t(z_1), \sigma_t^2(z_1))$, with $\mu_t(x) = tx_1$ and $\sigma_t(x) = 1 - (1 - \sigma_{\text{min}})t$. With these choices, the corresponding target vector field is given by

$$u_t(z \mid z_1) = \frac{z_1 - (1 - \sigma_{\text{min}})z}{1 - (1 - \sigma_{\text{min}})t}. \tag{5.6}$$

Sampling from the learned model can be obtained by first sampling $z_0 \sim \mathcal{N}(z \mid 0, 1)$ and then numerically solving eq. (5.2) for $z_1 = \phi_1(z_0)$.

5.2.3 Video Prediction

We introduce the main steps to train and use RIVER. First, as described in sec. 5.2.1 we use a per-frame perceptual autoencoder to reduce the dimensionality of data. Since the encoding is per-frame and thus the reconstruction error could be temporally inconsistent, we improve the quality of a generated video by also introducing an optional small autoregressive refinement step in the decoding network. Second, we train a denoising model via flow matching in the space of encoded frames with our distributed conditioning. Moreover, we accelerate the video generation by introducing a warm-start sampling procedure.

Training. We adapt Flow Matching [164] to video prediction by letting the learned vector field v_t condition on the past context frames. Furthermore, we randomize the conditioning

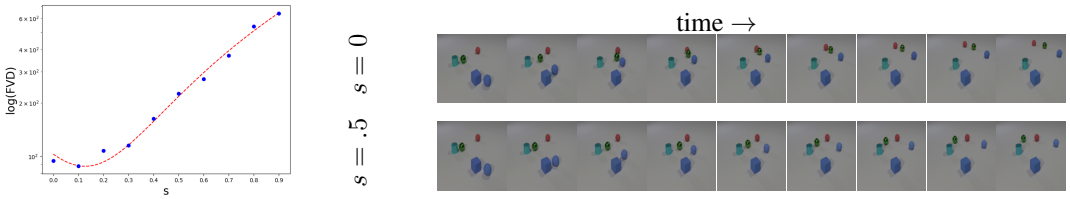


Figure 5.3: **Effect of warm-start sampling on the quality of generated frames.** *Left:* Warm-start sampling effect on the generation quality. Higher values of s for warm-start sampling lead to faster sampling, but worse FVD on the BAIR dataset [72]. Interestingly, $s = 0.1$ acts like the truncation trick [23, 178] and slightly improves the FVD. *Right:* The effect of extreme ($s = 0.5$) warm-start sampling strength which leads to reduced motion magnitude, The first frame in each row can be played as a video in Acrobat Reader.

at each denoising step to only 2 frames. This results in a very simple training procedure, which is described in Algorithm 1. Given a training video $\mathbf{z} = \{z^1, \dots, z^m\}$ (pre-encoded with VQGAN), we randomly sample a target frame z^τ and a random (diffusion) timestep $t \sim U[0, 1]$. We can then draw a sample from the conditional probability distribution $z \sim p_t(z \mid z^\tau)$ and calculate the target vector field $u_t(z \mid z^\tau)$ using eq. (5.6). We then sample another index c uniformly from $\{1, \dots, \tau - 2\}$ and use z^c, which we call *context frame*, together with $z^{\tau-1}$, which we call *reference frame*, as the two conditioning frames. Later, we show that the use of the reference is crucial for the network to learn the scene motion, since one context frame carries very little information about such motion. The vector field regressor v_t is then trained to minimize the following objective

$$\mathcal{L}_{\text{FM}}(\theta) = \|v_t(z \mid z^{\tau-1}, z^c, \tau - c\,; \theta) - u_t(z \mid z^\tau)\|^2, \tag{5.8}$$

where θ are the parameters of the model. Note that at no point during the training the whole video sequence must be stored or processed. Moreover, the generation of frames is never needed, which further simplifies the training process.

Inference. At inference time, in order to generate the T-th frame, we start from sampling an initial estimate z_0^T from the standard normal distribution (see Figure 5.2). We then use an ODE solver to integrate the learned vector field along the time interval $[0, 1]$. At each integration step, the ODE solver queries the network for $v_t(z_t^T \mid z^{T-1}, z^c, T - c)$, where $c \sim U\{1, \dots, T - 2\}$. In the simplest case, the Euler step of the ODE integration takes the form

$$z_{t_{i+1}}^T = z_{t_i}^T + \frac{1}{N} v_{t_i}(z_{t_i}^T \mid z^{T-1}, z^{c_i}, T - c_i), \tag{5.9}$$

$$c_i \sim U\{1, \dots, T - 2\}, \tag{5.10}$$

$$z_{t_0}^T \sim \mathcal{N}(z \mid 0, 1), \tag{5.11}$$

$$t_i = \frac{i}{N},\ i \in \{0, \dots, N - 1\}, \tag{5.12}$$

where N is the number of integration steps. We then use z_1^T as an estimate of z^T.

Refinement. When using a per-frame VQGAN [75], the autoencoded videos may not always be temporally consistent. To address this issue without incurring a significant computational cost, we optionally utilize a refinement network that operates in the pixel space. This deep convolutional network, based on the architecture of RCAN [294], is trained using the previous frame and the decoded next frame to refine the second frame. We train the model using an L_2 and a perceptual loss by refining 16 consecutive frames independently and then by feeding all frames to a perceptual network (I3D [30] in our case). We train the refinement network separately after training the autoencoder.

Sampling Speed. A common issue of models based on denoising processes is the sampling speed, as the same denoising network is queried multiple times along the denoising path in order to generate an image. This is even more apparent for the video domain, where the generation speed scales with the number of frames to generate. Some video diffusion models [107, 252] overcome this issue by sampling multiple frames at a time. However, the price they have to pay is the inability to generate arbitrarily long videos. We instead leverage the temporal smoothness of videos, that is, the fact that subsequent frames in a video do not differ much. This allows us to use a noisy previous frame as the initial condition of the ODE instead of pure noise. More precisely, instead of starting the integration from $z_0 \sim \mathcal{N}(z \mid 0, 1)$, we start at $z'_s \sim p_s(z \mid z^{T-1})$, where $1 - s$ is the speed up factor. We call this technique *warm-start sampling*.

Intuitively, larger s results in a lower variability in the future frames. Moreover, we found that starting closer to the end of the integration path reduces the magnitude of the motion in the generated videos (see Figure 5.3 (right)), since the model is required to sample closer to the previous frame. Therefore, there is a tradeoff between the sampling speed and the quality of the samples. We further emphasize this tradeoff by computing the FVD [246] of the generated videos depending on the speed up factor $1 - s$ (see Figure 5.3 (left)).

5.2.4 Implementation

A commonly leveraged architecture for flow matching and diffusion models is UNet [217]. However, we found that training UNet could be time demanding. Instead, we propose to model $v_t(z \mid z^{\tau-1}, z^c, \tau - c\,; \theta)$ with the recently introduced U-ViT [15]. U-ViT follows the standard ViT [67] architecture and adds several long skip-connections, like in UNet. This design choice allows U-ViT to achieve on par or better results than UNet on image generation benchmarks with score-based diffusion models.

The inputs to the network are ${}^{HW}/_{f^2}$ tokens constructed by concatenating $z, z^{\tau-1}$ and z^c in feature axis as well as one additional time embedding token t that makes the network time-dependent. We additionally add spatial position encondings to the image tokens and augment $z^{\tau-1}$ and z^c with an encoded relative distance $\tau - c$ to let the network know how far in the past the condition is. That is, the overall input to the network is of size $[{}^{HW}/_{f^2} + 1, 3 \times d]$,

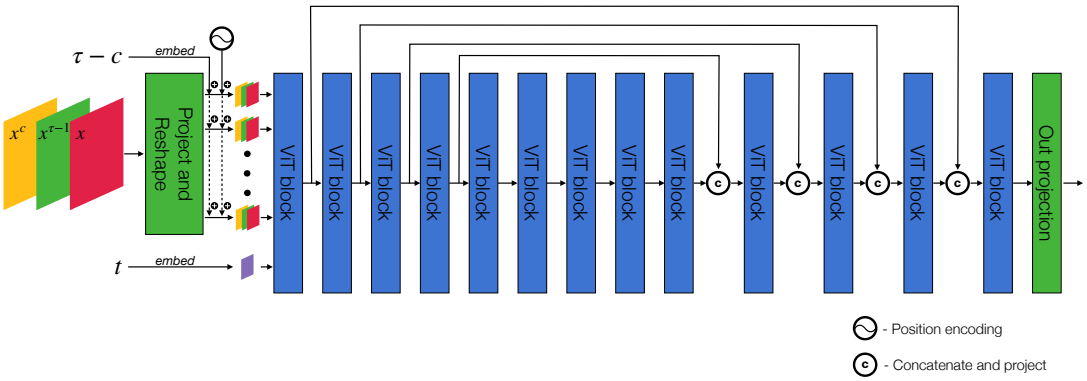


Figure 5.4: **Architecture of the vector field regressor of RIVER.** "ViT block" stands for a standard self-attention block used in ViT [67], that is an MHSA layer, followed by a 2-layer wide MLP, with a layer normalization before each block and a skip connection after each block. "Out projection" involves a linear layer, followed by a GELU [112] activation, layer normalization and a 3×3 convolutional layer.

Table 5.1: ***KTH* dataset quantitative evaluations.** The evaluation protocol is to predict the next 30/40 frames given the first 10 frames.

Setting	Method	FVD↓	PSNR↑	SSIM↑
10→30	SRVP [89]	222	29.7	**0.87**
	SLAMP [2]	228	29.4	**0.87**
	MCVD [252]	323	27.5	0.84
	RIVER (ours)	**180**	**30.4**	0.86
10→40	MCVD [252]	276.7	26.4	0.81
	GridKeypoints [92]	**144.2**	27.1	**0.84**
	RIVER (ours)	170.5	**29.0**	0.82

where the first dimension refers to the number of tokens, while the second refers to the number of channels (see Figure 5.4).

5.3 Experiments

In section 5.3.1, we report our results on several video prediction benchmarks. We evaluate our method using standard metrics, such as FVD [246], PSNR and SSIM [261]. We additionally show in section 5.3.2 that our model is able to perform visual planning. Video generation is demonstrated in section 5.3.3. Note that if not explicitly specified, we use the model without the refinement stage and with $s = 0$ in warm-start sampling.

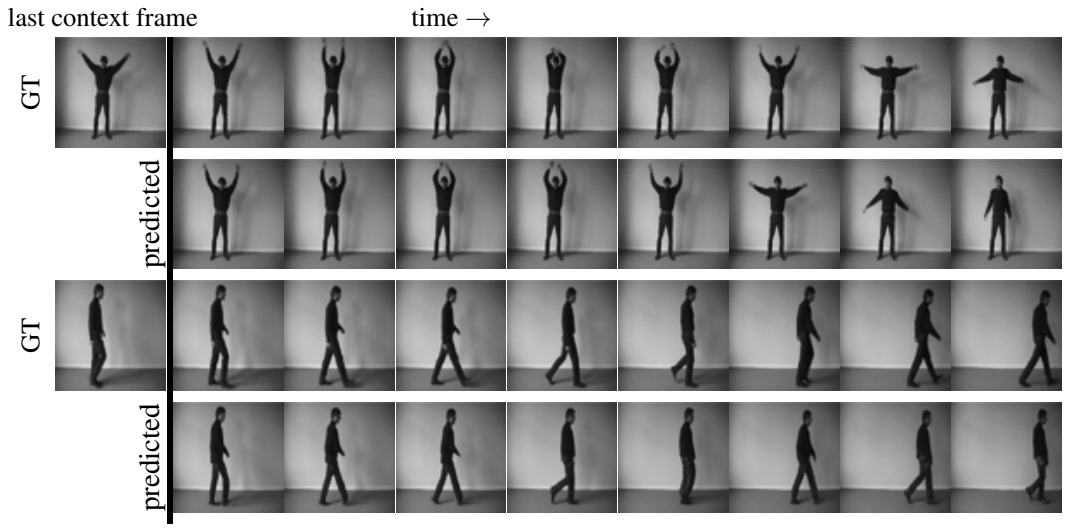


Figure 5.5: **Video prediction on the *KTH* dataset.** In order to predict the future frames, the model conditions on the first 10 (context) frames. Of this sequence, only the last context frame is shown.

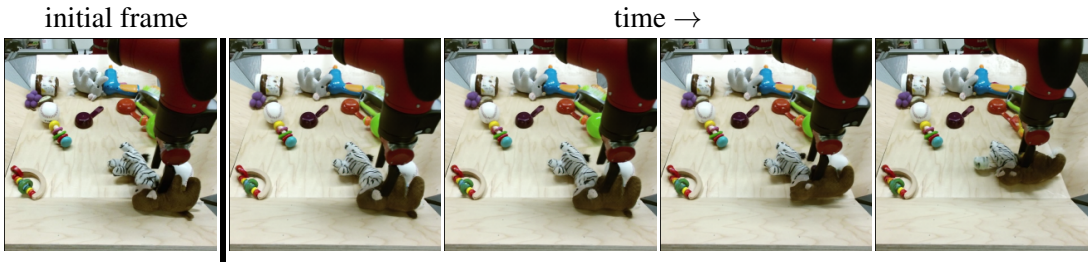


Figure 5.6: **Video prediction on the *BAIR* dataset.** The model predicts future frames conditioned on a single initial frame. Thanks to VQGAN, RIVER can be used to generate high resolution videos.

5.3.1 Conditional Video Prediction

We test our method on 2 datasets. First, to assess the ability of RIVER to generate structured human motion, we test it on the **KTH** dataset [225]. KTH is a dataset containing 6 different human actions performed by 25 subjects in different scenarios. We follow the standard evaluation protocol predicting 30/40 future frames conditioned on the first 10 at a 64×64 pixel resolution. The results are reported in Table 5.1. We show that RIVER achieves state of the art prediction quality compared to prior methods that do not use domain-specific help. For instance, [92] models the motion of the keypoints, which works well for human-centric data, but does not apply to general video generation. Figure 5.5 shows qualitative results.

Table 5.2: ***BAIR* dataset evaluation.** We follow the standard evaluation protocol, which is to predict 15 future frames given 1 initial frame. The common way to compute the FVD is to compare 100×256 generated sequences to 256 randomly sampled test videos. Additionally, we report the numbers of the network without the refinement stage versus the original ground truth (RIVER *w/o refine*) and the autoencoded ground truth (RIVER *w/o refine vs ae GT*) to highlight the influence of the VQGAN's artifacts on the assessment of the motion consistency.

Method	FVD↓	Mem (GB)	Training Hours
TriVD-GAN-FP [175]	103.0	1024	280
Video Transformer [264] (L)	94.0	512	336
CCVS [149] (low res)	99.0	128	40
CCVS [149] (high res)	80.0	-	-
LVT [210] ($n_c = 4$)	125.8	128	48
FitVid [11]	93.6	1024	288
MaskViT [103]	93.7	-	-
MCVD [252] (concat)	98.8	77	78
MCVD [252] (spatin)	103.8	86	50
NÜWA [270]	86.9	2560	336
RaMViD [119]	84.2	320	72
VDM [117]	66.9	-	-
RIVER *w/ refine*	106.1	25	25
RIVER *w/o refine*	145.8	-	-
RIVER *w/o refine vs ae GT*	73.5	-	-

Additionally, in Table 5.2 we evaluate the capability of RIVER to model complex interactions on **BAIR** [72], which is a dataset containing around 44K clips of a robot arm pushing toys on a flat square table. For BAIR, we generate and refine 15 future frames conditioned on one initial frame at a 64×64 pixel resolution. Due to the high stochasticity of motion in the BAIR dataset, the standard evaluation protocol in [11] is to calculate the metrics by comparing 100×256 samples to 256 random test videos (*i.e.*, 100 generated videos for each test video, by starting from the same initial frame as the test example). Additionally, we report the compute (memory in GB and hours) needed to train the models. RIVER reaches a tradeoff between the FVD and the compute and generates smooth realistic videos while requiring much less computational effort (see also Figure 5.1). In addition, we calculate the FVD vs the autoencoded test set, as we find that FVD (like FID [200]) can be affected even by different interpolation techniques. This way we eliminate the influence of potential autoencoding artifacts on the metrics in order to assess the consistency of the motion only. In fact, there is an improvement of about 30% in the FVD. Furthermore, although the standard benchmark on BAIR uses 64×64 pixels resolution, with the help of the perceptual compression, we are able to generate

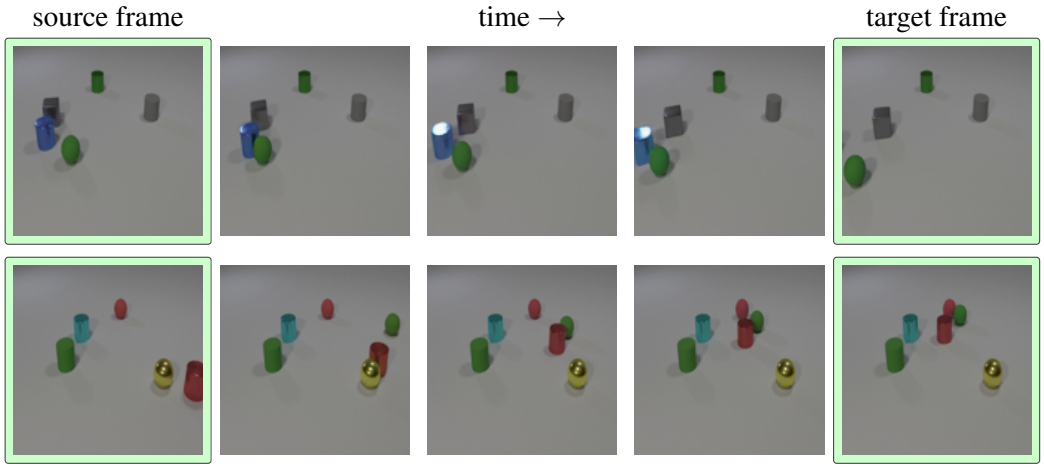


Figure 5.7: **Visual planning with RIVER on the *CLEVRER* dataset.** Given the source and the target frames, RIVER infills the frames in between. Note how the model manipulates the objects by forcing them to interact in order to achieve the goal. In some cases this even requires introducing new objects into the scene.

higher-resolution videos under the same training costs. See Figure 5.6 for qualitative results on the *BAIR* dataset at 256×256 resolution.

5.3.2 Visual Planning

One way to show the ability of the model to learn the dynamics of the environment is to do planning [86, 87, 276]. With a small change to the training of our model, RIVER is able to infill the video frames given the source and the target images. The only change to be done to the model is to remove the reference frame and to let two condition frames be sampled from both the future frames and the past ones. At inference time, given the source and the target frames, our model sequentially infills the frames between those. We show in Figure 5.7 some qualitative results of video interpolation on the CLEVRER [283] dataset, which is a dataset containing 10K training clips capturing a synthetic scene with multiple objects interacting with each other through collisions. It is a dataset suitable for planning, as it allows to show the ability of the method to model the dynamics of the separate objects and their interactions. We test our model at the 128×128 pixels resolution. Note how the model has learned the interactions between the objects and is able to manipulate the objects in order to achieve the given goals.

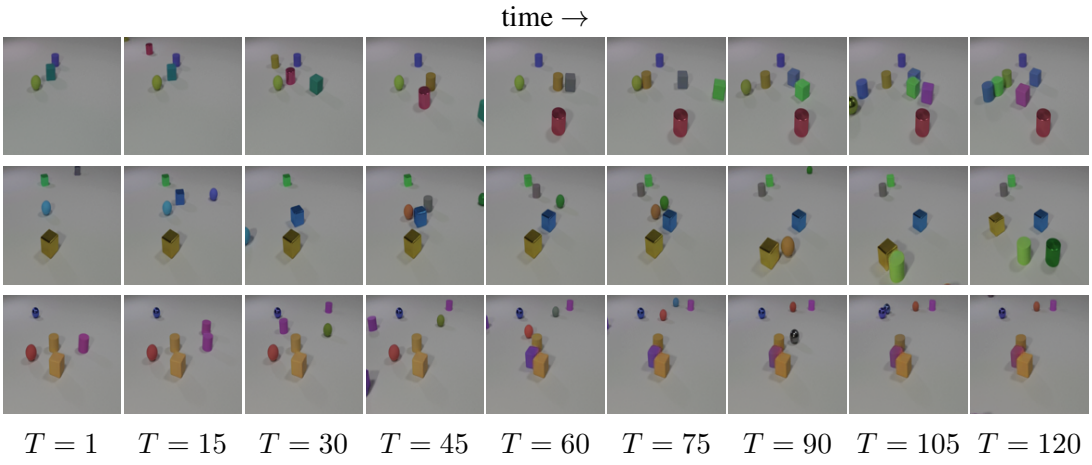


Figure 5.8: **Long video generation on the *CLEVRER* dataset.** We generate the first frame and predict the next frames.

5.3.3 Video Generation

RIVER can be easily adapted to support *video generation*. Inspired by the classifier-free guidance [114] we train a single model to both generate (the first frame of a video) and predict the next frames by simply feeding noise instead of the condition frames 10% of the times during training. Then, during inference we generate the first frame and then predict the rest of the video given the first frame. Figure 5.8 shows our results for video generation on CLEVRER [283] (FVD $= 23.63$). Other methods [179, 231, 285] have difficulties in modeling the motions and interactions of objects. For videos and qualitative comparisons, visit our website[1].

5.3.4 Ablations

In this section, we ablate several design choices in order to illustrate their impact on the performance of RIVER.

First, we ablate the importance of using the reference frame in the condition. In [262], where the stochastic conditioning was first introduced, only one view from the memory was used at each denoising step for generating a novel view. However, conditioning on one frame from the past does not work for video prediction, since one frame does not contain any information about pre-existing motion. We train a model, where we remove the reference frame from the condition and compare its performance to the full model. For this ablation we test RIVER on the CLEVRER [283] dataset. We found that without the reference frame in the condition the model is confused about the direction of the motion, which results in jumping objects (see Figure 5.9). For the quantitative results, check Table 5.3.

[1] https://araachie.github.io/river

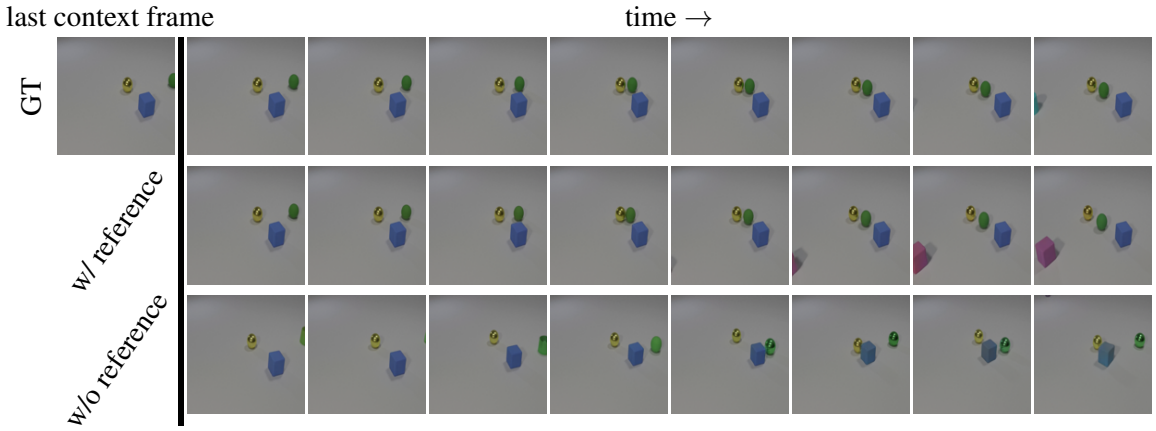


Figure 5.9: **Video prediction on the *CLEVRER* dataset.** The model trained with two frames is consistent, while the model w/o reference changes the type of green object and does not model motion correctly. The green object hits the blue cube and then comes back to hit it again (last frames of the picture).

Given a model trained so that the context frames are sampled from the whole past of a sequence, at inference time we ablate the size of the past window used for the context frames to better understand the impact of the history on the video generation performance. In this ablation, we uniformly sample the context frames from $\{\tau - 1 - k, \dots, \tau - 2\}$ for $k = 2, 4, 6, 8$, and show which past frames better support RIVER's predictions. For this experiment we use our trained model on the BAIR [72] and KTH [225] datasets. Since there are occlusions in BAIR, we suspect that having more context can help to predict the future frames more accurately. Having more context frames also helps to predict a smoother motion for humans in KTH. Table 5.4 shows that there is a trade-off in context size and although having more context can be useful, on simple datasets having only a few frames is better to solve the prediction task.

Finally we show in Figure 5.3 (left) that warm-start sampling can be used to generate samples faster (with fewer integration steps) but with a cost on quality. Interestingly we observed that a small speed up factor actually helps the sampling and despite having fewer integration steps leads to better performance. We suspect that this effect is similar to the truncation trick [23, 178] in GANs. Notice however, that compared to other diffusion-based video generation approaches, RIVER conditions only on 2 past frames for a single neural function evaluation (NFE). Hence, a single NFE is generally less expensive. For instance, it takes 9.97 seconds for RIVER to generate 16 frames video, while RaMViD [119] requires 40.47 seconds with a vanilla scheduler on a single Nvidia GeForce RTX 3090 GPU (on BAIR with 64×64 resolution). For more results, see the supplementary material.

For the CLEVRER [283] dataset, we implemented random color jittering as an additional data augmentation technique. This step was crucial in preventing overfitting, as we observed

Table 5.3: **Ablations on the use of the reference frame.** We generate 14 frames given 2 initial ones and the metrics are calculated on 256 test videos with 1 sample per video and 10 integration steps per frame. All models are trained for 80K iterations.

Method	FVD↓	PSNR↑
w/ reference	94.38	30.53
w/o reference	217.13	26.95

Table 5.4: **Ablations on the context size.** Using a pretrained model on BAIR [72] and KTH [225] we observe a trade-off wrt the number of conditioning frames. We believe that datasets with more challenging scenes and dynamics may require more context frames.

Context	BAIR / PSNR↑	KTH / PSNR↑
2 frames	25.64	28.53
4 frames	25.94	29.07
6 frames	26.00	30.17
8 frames	25.28	29.40

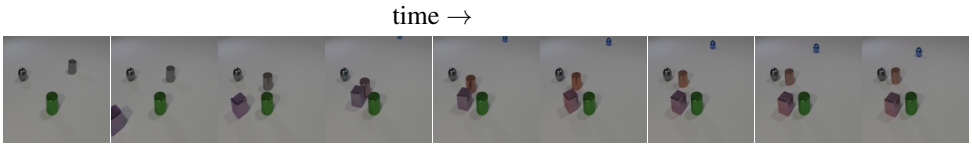


Figure 5.10: **Color change in CLEVRER.** A sequence generated with RIVER trained on the *CLEVRER* dataset without data augmentation. Notice how the color of the grey cylinder changes after its interaction with the cube. In order to prevent such behaviour, both the autoencoder and RIVER are trained with random color jittering as data augmentation. The first frame can be played as a video in Acrobat Reader.

that without it, object colors tended to change inconsistently in the generated video sequences (see Figure 5.10).

In Figure 5.11 we show the FVD [246] and PSNR of RIVER trained on CLEVRER [283] against the iteration time. As we can see, the training is stable and more iterations lead to better results.

We conducted an ablation study comparing Flow Matching (FM) to Denoising Diffusion Probabilistic Models (DDPM). Our qualitative results demonstrate that DDPM fails to converge on the BAIR dataset, while FM shows successful convergence. This performance difference, coupled with FM's faster convergence rate, motivated our choice of FM for our method (see Figure 5.12).

In Table 5.5, we compare the total training time and GPU (or TPU) memory requirements of different models trained on BAIR64×64 [72]. As we can see, RIVER is extremely efficient and can achieve a reasonable FVD [246] with significantly less compute than the other methods. For example, SAVP [151], which has the same FVD as RIVER, requires 4.6× more compute

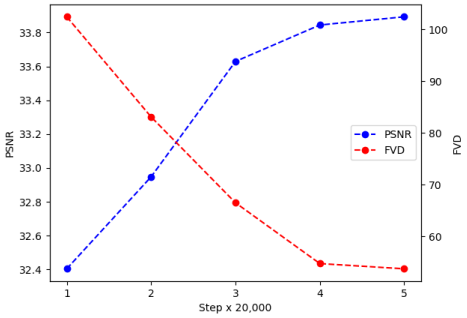


Figure 5.11: **Training curve of RIVER on CLEVRER.** As we can see, the training is stable and more iterations lead to better results, both in terms of PSNR and FVD.

FM DDPM

Figure 5.12: **Video generation quality difference between FM and DDPM.** DDPM fails to converge on BAIR dataset with the exact same hyperparameters as RIVER which is based on Flow Matching. Use Acrobat Reader to play videos.

(measured by Mem×Time) and all the models that take less compute than RIVER have FVDs more than 250.

5.4 Discussion

In this chapter, we introduced RIVER, a novel approach to video prediction that extends the principles of sparsity and efficient processing to generative tasks. Building upon the concepts explored in previous chapters, RIVER demonstrates how strategic sparsity can be applied to the challenging domain of video synthesis and prediction, serving as an implicit world model.

RIVER's adaptation of Flow Matching to video data, coupled with its innovative use of randomly and sparsely sampled context frames, addresses the fundamental challenge of efficiently processing and generating high-dimensional temporal data. This approach allows for conditioning on an arbitrarily large window of past frames, enhancing the model's ability to capture long-term dependencies while maintaining computational efficiency. This aligns with our thesis's broader goal of leveraging sparsity to unlock new capabilities in model training and application, particularly in the context of world modeling through video prediction.

The extensive experiments across various video datasets validate RIVER's effectiveness not only in predicting high-quality videos but also in its flexibility to perform related tasks such as visual planning and video generation. This versatility underscores the potential of sparse, efficient processing techniques in addressing a wide range of video-related challenges, and highlights RIVER's capability as an implicit world model.

As we transition to the next chapter on ViDROP, we extend the principles of sparsity and efficient processing explored in RIVER to the domain of video understanding tasks. This

Table 5.5: **Detailed training compute comparisons.** We report the memory and training times requirements of different models trained on BAIR64×64 [72]. The overall compute (Mem × Time) shows that RIVER delivers better FVD with much less compute.

Method	VRAM (GB)	Time (Hours)	**Mem×Time (GB×Hour)**	FVD [246]
RVD [282]	24	-	-	1272
MoCoGAN [245]	16	23	368	503
SVG-FP [57]	12	24	288	315
CDNA [87]	10	20	200	297
SV2P [10]	16	48	768	263
SRVP [89]	36	168	6048	181
VideoFlow [148]	128	336	43008	131
LVT [210]	128	48	6144	126
SAVP [151]	32	144	4608	116
DVD-GAN-FP [45]	2048	24	49152	110
Video Transformer(S) [264]	256	33	8448	106
TriVD-GAN-FP [175]	1024	280	286720	103
CCVS(Low res) [149]	128	40	5120	99
MCVD(spatin) [252]	86	50	4300	97
Video Transformer(L) [264]	512	336	172032	94
FitVid [11]	1024	288	294912	94
MCVD(concat) [252]	77	78	6006	90
NUWA [270]	2560	336	860160	87
RaMViD [119]	320	72	23040	83
RIVER	25	25	625	106

progression allows us to investigate how these techniques can be applied to create dense video representations, further demonstrating the versatility of our approach in tackling various aspects of video analysis and processing.

5.5 Extra Qualitative Results

Here we provide more visual examples of the sequences generated with RIVER. See Figures 5.14 and 5.16 for results on the BAIR [72] dataset, Figures 5.13 and 5.15 for results on the KTH [225] dataset and Figures 5.17 and 5.19 for video prediction and planning on the CLEVRER [283] dataset respectively. Besides this, we highlight the stochastic nature of the generation process with RIVER in Figure 5.18.

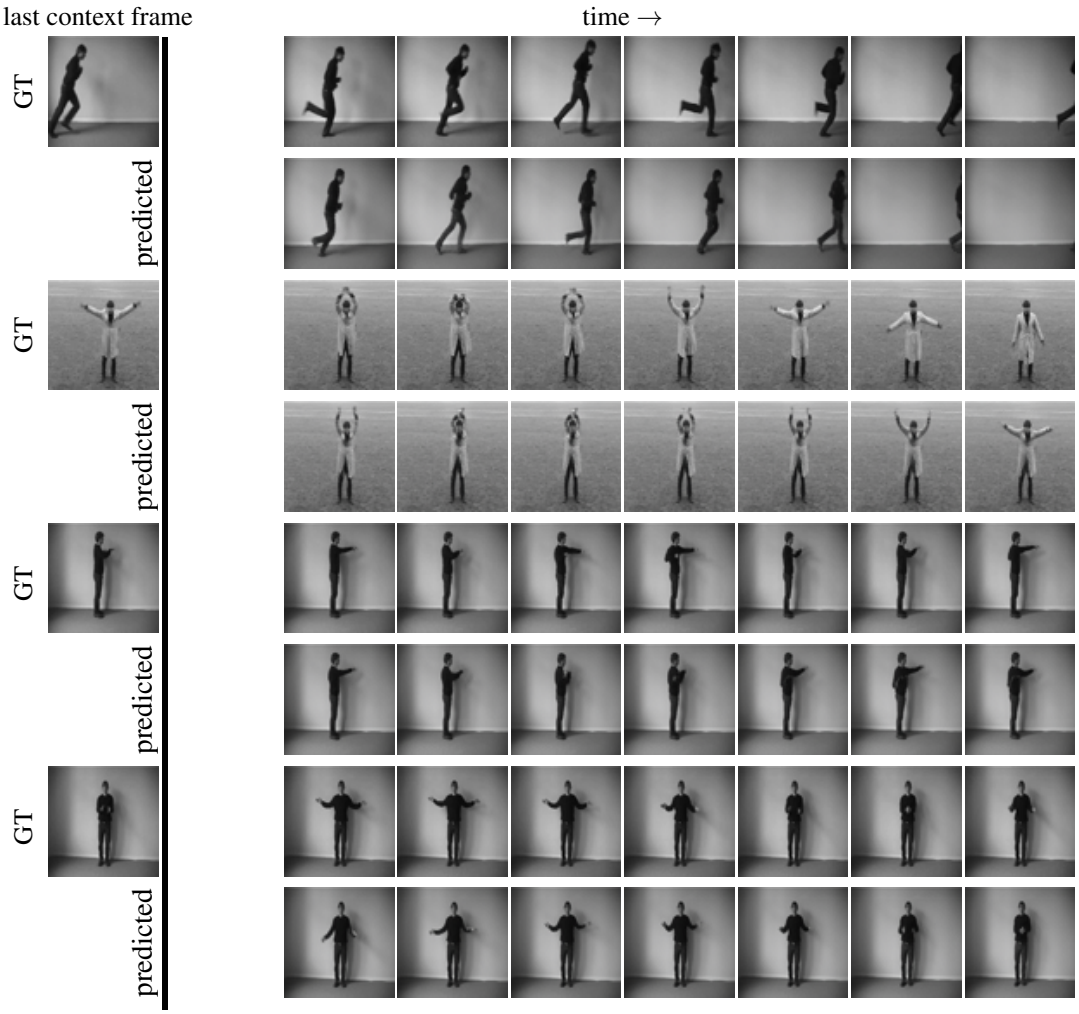


Figure 5.13: **Extra video prediction samples on the *KTH* dataset.** Odd rows show frames of the original video. Even rows show the video generated by RIVER when fed the context frames of the row above (GT). We observe that RIVER is able to generate sequences with diversity and realism. The images in the first column after the bold vertical line can be played as videos in Acrobat Reader.

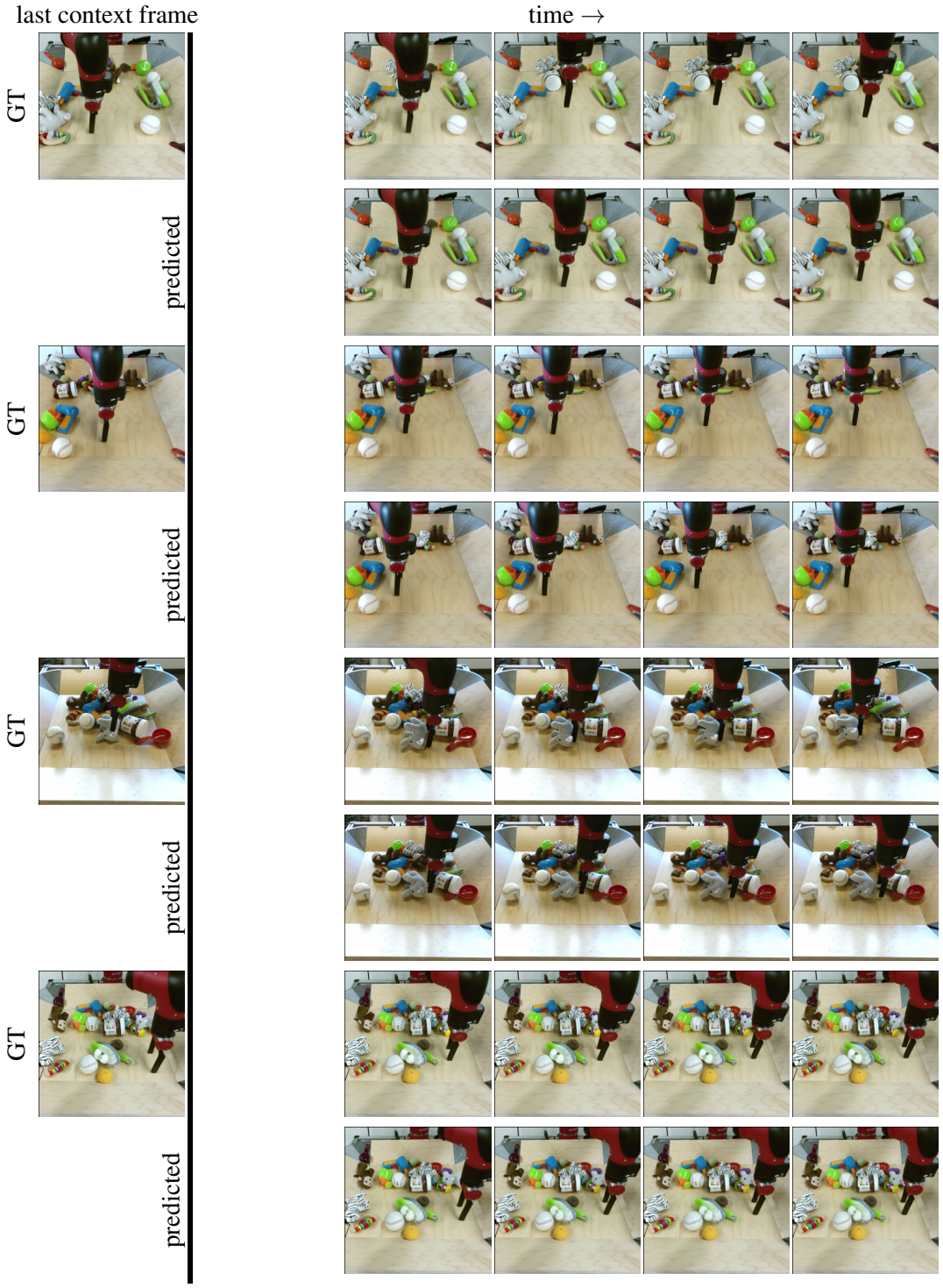


Figure 5.14: **Extra video prediction samples on the *BAIR* dataset at** 256×256 **resolution.** The model predicts the future frames conditioned on a single initial frame. The frames in the first column after the bold vertical line can be played as videos in Acrobat Reader.

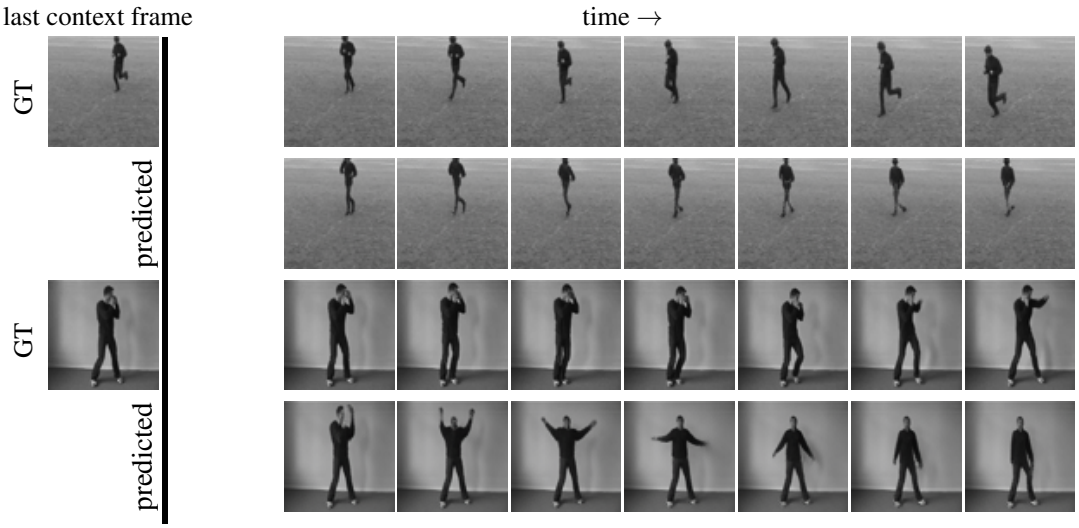


Figure 5.15: **Failure cases on the *KTH* dataset.** A common failure mode is when a certain action gets confused with another one, which results in a motion that morphs into a different one. In all examples the model is asked to predict 25 future frames given the first 5. The images in the first column after the bold vertical line can be played as videos in Acrobat Reader.

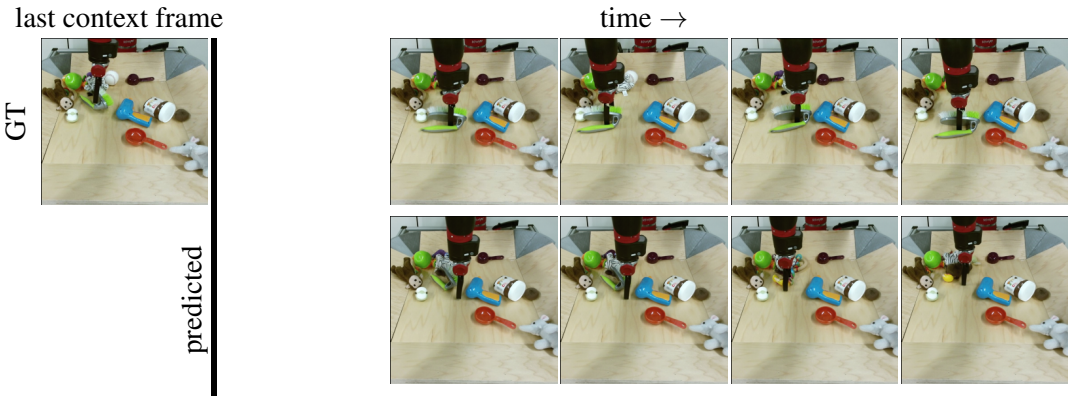


Figure 5.16: **Failure case on the *BAIR* dataset.** A common failure mode emerges when generating longer sequences and is when the interaction causes objects to change their class, shape or even to disappear. The images in the first column after the bold vertical line can be played as videos in Acrobat Reader.

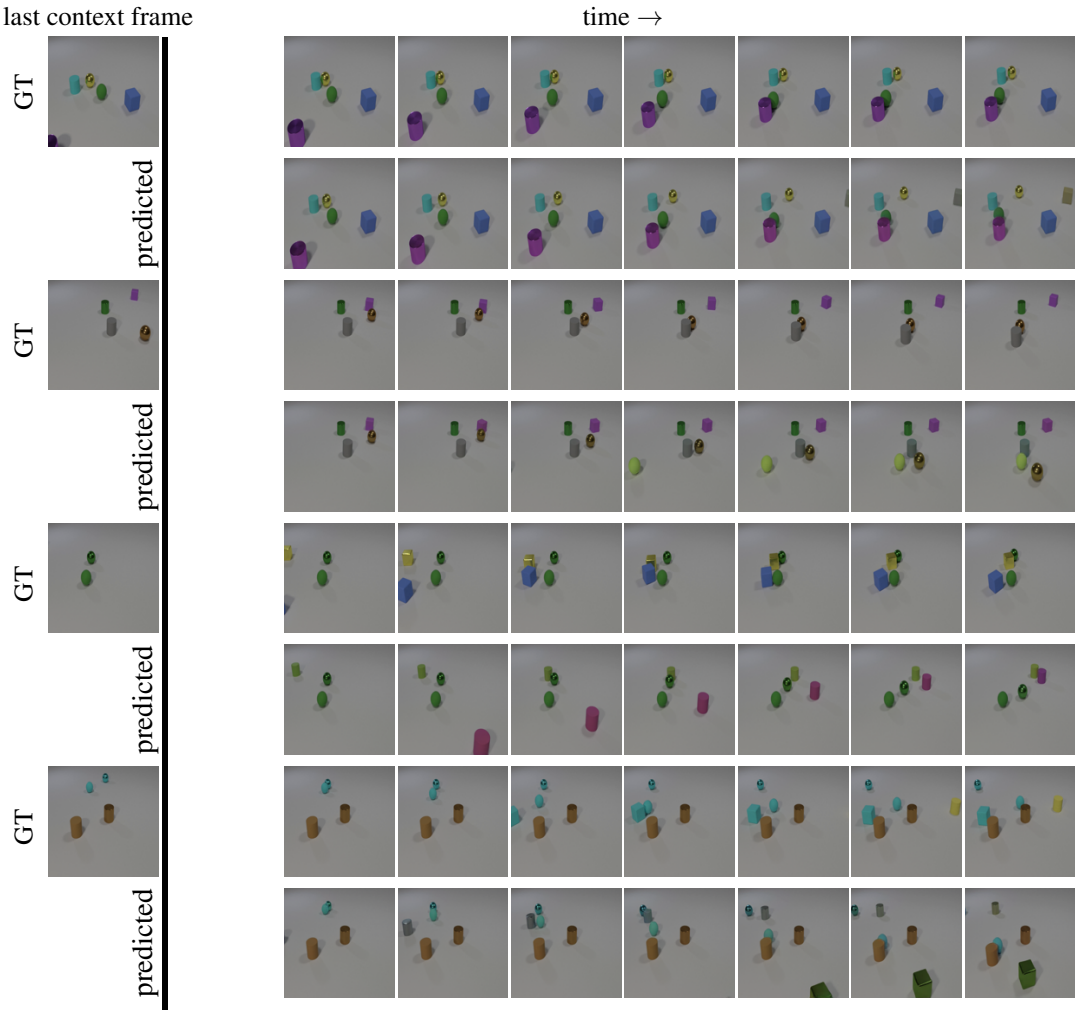


Figure 5.17: **Extra video prediction samples on the *CLEVRER* dataset.** In order to predict the future frames, the model conditions on the first 2 frames. Only the last context frame is shown. The model succeeds to predict the motion that was observed in the context frames. However, it cannot predict new objects as in the ground truth and introduces random new objects due to the stochasticity of the generation process. The images in the first column after the bold vertical line can be played as videos in Acrobat Reader.

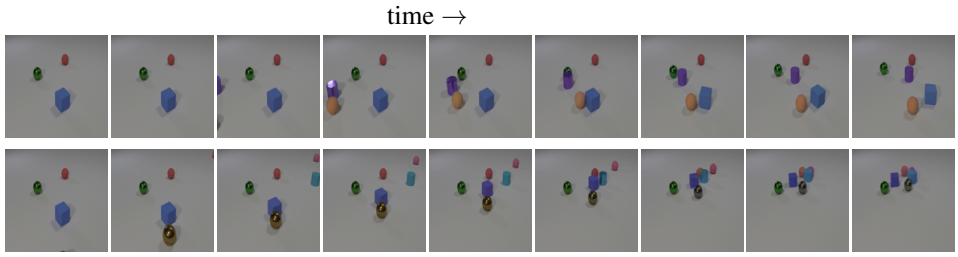


Figure 5.18: **Two sequences generated with RIVER trained on the *CLEVRER* dataset.** The model was asked to predict 19 frames given 1. Note the very different fates of the blue cube in these two sequences. The images in the first column can be played as videos in Acrobat Reader.

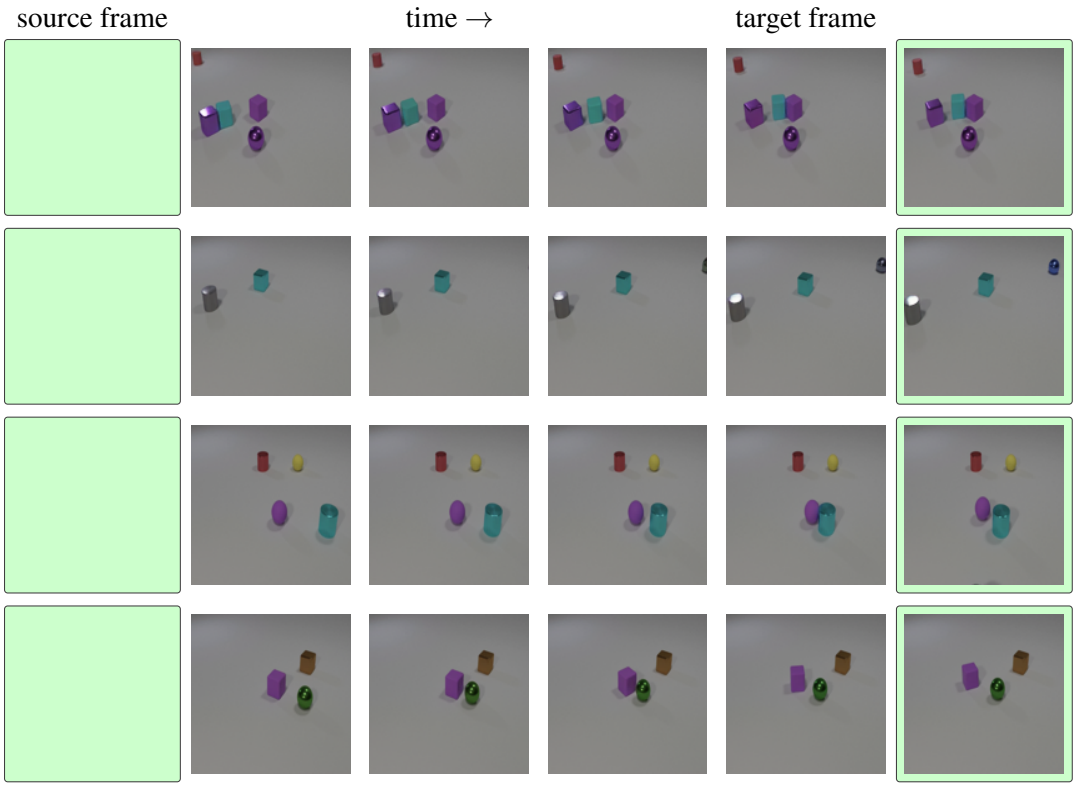


Figure 5.19: **Extra visual planning samples with RIVER on the *CLEVRER* dataset.** Given the source and the target frames, RIVER generates intermediate frames, so that they form a plausible realistic sequence. The images in the first column can be played as videos in Acrobat Reader.

Chapter 6

ViDROP: Video Dense Representation through Omissive Processing

Material in this chapter is based on: Sameni, Sepehr, Simon Jenni and Paolo Favaro. "ViDROP: Video Dense Representation through Omissive Processing." Manuscript in preparation, 2024.

Building upon our exploration of sparsity and efficient processing in self-supervised learning (SSL) from the previous chapters, we now address a critical challenge in video understanding: the prohibitive costs associated with fine-tuning large-scale models. This chapter introduces ViDROP (Video Dense Representation through Omissive Processing), a novel approach that strategically combines generative (reconstructive) and discriminative learning paradigms. By integrating these complementary techniques, ViDROP aims to bridge the gap between the rich representations learned by generative models and the task-specific prowess of discriminative approaches, all while significantly reducing the computational burden of fine-tuning. This synthesis not only enhances model capabilities in video understanding tasks but also demonstrates how strategic sparsity and efficient processing can lead to more adaptable and resource-efficient video analysis frameworks.

Recent advancements in SSL, exemplified by models like DINOv2 [196], have pushed the boundaries of image understanding to new heights. A notable feature of DINOv2 is its incorporation of a dense (per-patch) self-distillation loss alongside a global consistency loss, capturing both localized features and holistic understanding. However, extending these techniques to video presents substantial challenges due to the increased dimensionality and data complexity of video sequences.

Previous attempts to address these challenges, such as VideoMAE [244] and V-JEPA [18], have employed sparse encoder and dense decoder architectures. While promising, these approaches face limitations in computational cost and the need for fine-tuning on dense inputs for downstream tasks. ViDROP overcomes these limitations by uniquely combining

token dropping and masking strategies within a single encoder architecture, offering both computational efficiency and rich representational learning.

Our approach builds upon the insights gained from DILEMMA, SCALE, and RIVER, leveraging strategic sparsity to enhance model capabilities while reducing computational overhead. ViDROP extends these concepts by introducing a novel approach to video compression and efficient processing, enabling state-of-the-art performance across various video understanding tasks without relying on heavy data augmentations.

Inspired by the successes of SVT [212] and SCALE in adapting SSL models to new domains, we initialize ViDROP with pretrained weights from established models. We adapt these techniques within the DINOv2 framework to effectively handle video data, maintaining the rich representational power of the pretrained models while optimizing for our novel objective.

To address the bottleneck of video data loading and preprocessing, particularly crucial for efficient experimentation with smaller models, we introduce a simple yet effective compression technique using k-means clustering [172] for frame patches. This approach not only enhances efficiency but also keeps our representations close to the pixel space, allowing us to leverage pretrained checkpoints from large-scale models trained on pixel data.

Our contributions can be summarized as follows:

- We introduce ViDROP, a novel architecture combining sparse token processing with masked learning, enabling DINOv2-like performance for video data without the associated training computational burden.

- We propose an effective video compression technique using k-means clustering in pixel space, significantly accelerating the training process while maintaining compatibility with pretrained models.

- We demonstrate the scalability of our approach by extending it to larger networks and training models from scratch, showcasing its general applicability across different network sizes and training regimes.

- We achieve state-of-the-art performance on various video understanding benchmarks, including action classification and temporal action detection tasks, confirming the efficacy of our approach in a domain dominated by fine-tuned SSL models.

By focusing on efficiency and scalability in video understanding, ViDROP not only advances our understanding of SSL for video representation but also aligns with the broader goals of this thesis in exploring how strategic sparsity can unlock new capabilities in model training and application. The subsequent sections will dive deeper into the methodology, experiments, and results, illustrating how ViDROP contributes to the evolving landscape of efficient and effective self-supervised learning in video analysis.

6.1 Background

This chapter builds upon the foundations of self-supervised learning (SSL) for video representations, as detailed in Section 2.2. ViDROP addresses several key challenges in video SSL that have been highlighted throughout this thesis, with a particular focus on efficiency and computational considerations.

A critical concept in ViDROP's approach is the extension of token dropping techniques, discussed in Section 2.2.5, to create a more computationally efficient architecture for video understanding tasks. By eliminating the need for a complex decoder structure, ViDROP tackles the computational challenges associated with processing high-dimensional video data, a key issue in scaling video SSL models.

Another important aspect of ViDROP is its approach to multi-view learning, building on the concepts outlined in Section 2.2.4. Our method leverages full spatio-temporal attention to capture and fuse features at different spatio-temporal resolutions, aiming to capture rich temporal dependencies while maintaining computational efficiency. This approach aligns with state-of-the-art video models while addressing the efficiency concerns central to this thesis.

Lastly, ViDROP introduces a novel k-means clustering-based video compression technique to address the data bottleneck challenges mentioned in Section 2.2.5. This approach, not previously elaborated in the main background chapter, enables faster data loading and experimentation while keeping representations close to the pixel space. By doing so, ViDROP advances the thesis's focus on efficient processing and representation learning in video SSL.

6.2 Method

ViDROP (Video Dense Representation through Omissive Processing) introduces a novel approach to self-supervised learning for video understanding, combining the strengths of encoder-only and asymmetric models while addressing their limitations. Our method extends the DINOv2 [196] framework to video data, incorporating sparsity and adapting it for our compressed data format.

6.2.1 Video Sampling and Processing

Given a video, we sample multiple clips, creating two large crops and eight small crops. The large crops are fed directly to an exponential moving average (EMA) of the network, serving as the teacher. For the student network, we apply sparsification (token dropping) to these large crops and replace some of the remaining tokens with a special MSK token. Small crops maintain dense inputs without sparsification or masking [7]. Both student and teacher networks include a CLS token for global representation.

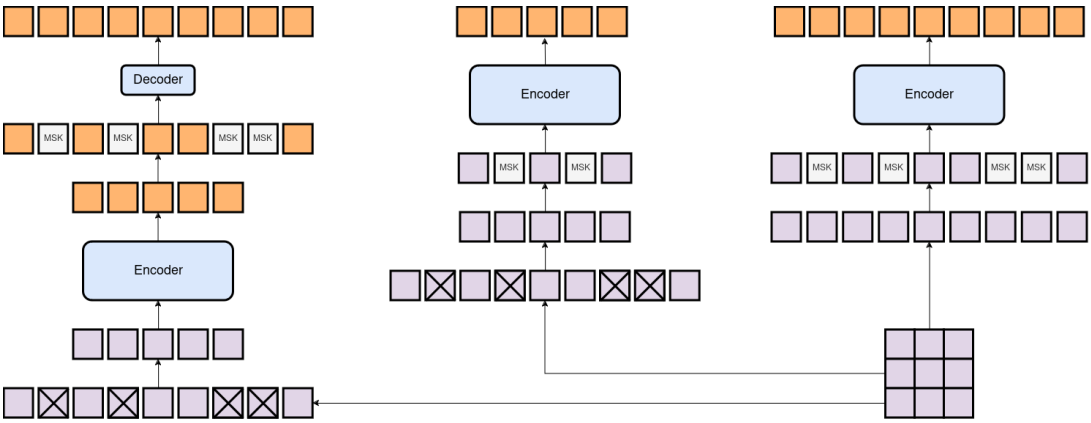


Figure 6.1: **Comparison of patch reconstruction architectures.** This figure illustrates three different approaches to patch reconstruction. *Left:* Traditional Encoder/Decoder architecture, where a sparse set of input patches is fed to the encoder, and a small decoder reconstructs all dropped tokens. *Middle:* Our proposed ViDROP architecture, which uniquely combines sparse (dropped) and masked (MSK) tokens in a single encoder. *Right:* Masked Encoder approach (such as DINOv2 [196]) that uses only MSK tokens. In all methods, the input is flattened and patched (image or video), and targets are either pixels or the output of the teacher network (an exponential moving average of the encoder, omitted for clarity).

6.2.2 Loss Function

ViDROP's loss function consists of two main components. The first is a DINO-style loss calculated between the CLS tokens of the student and teacher [29]. The second is a patch-level loss for the masked tokens in the student, using the corresponding outputs from the teacher as targets [299]. We employ a global-global consistency loss to ensure coherence between different views of the same video and a global-local consistency loss that aligns features from large crops (global) with those from small crops (local). The patch-level loss enables fine-grained representation learning, complemented by the KoLeo regularization adapted from DINOv2. For detailed formulations of these losses, we refer readers to the DINOv2 paper [196].

Unlike traditional consistency-based SSL approaches, we minimize the use of data augmentation [185], relying primarily on heavy masking and our lossy data compression technique as pseudo-augmentations.

6.2.3 Architecture

ViDROP employs a sparse encoder with masked tokens, eliminating the need for a separate decoder (see Figure 6.1). This design choice sets us apart from asymmetric models that rely on encoder-decoder architectures [8, 12, 18, 110, 167, 244]. Our architecture uniquely combines token dropping and masking strategies within a single encoder, achieving the computational

efficiency of sparse processing while maintaining the rich representational learning capabilities of mask-based self-distillation approaches [13, 16, 196, 279, 299].

6.2.4 Lossy Data Loading

To address the data loading bottleneck in video processing, we introduce a lossy data loading scheme based on k-means clustering [172] of video patches. By applying k-means clustering directly in pixel space on 10×10 pixel patches, we achieve a compression rate of 150, significantly accelerating the data loading process.

This approach offers several advantages over VQVAE-based methods [75, 198, 247, 266]. It allows for faster processing during both training and inference by avoiding complex encoding and decoding steps. Our method maintains compatibility with pretrained models that operate on raw pixel data, enabling us to leverage existing large-scale pretrained checkpoints. Furthermore, our approach provides a flexible compression scheme that can be easily adjusted (by changing the patch size) to balance between compression rate and representation quality.

6.3 Experiments

6.3.1 Experimental Setup and Protocols

We use the training set of Kinetics-400 [139] for self-supervised training. For evaluation, we use four common action classification datasets (Kinetics-400, Something-Something-v2 [101], UCF101 [233], and HMDB51 [147]) and THUMOS14 [135] for temporal action detection. For all evaluations, contrary to common settings in reconstruction-based models [13, 18, 110, 167, 244] but compatible with consistency-based models that rely on heavy data augmentations, we use a **frozen** backbone and only train a linear head on top (except in the case of THUMOS14, where we train a transformer following ActionFormer [291]).

Unless stated otherwise, our ViT-Base models are trained for 60 epochs using k-means compressed data (40 epochs for ViT-Large without compression) on 4 NVIDIA-4090 GPUs with a total batch size of 512 (with gradient accumulation). The training of Base models takes 24 hours and 57 hours for Large models. During training, 85% of the tokens are dropped for the student when fed with two large clips (16 frames of 224×224). For half of the mini-batch, we randomly mask a portion of the remaining tokens. Additionally, 8 local crops of size 96×96 (8 frames) are fed as dense input without token dropping or masking.

6.3.2 Ablations

Here we perform in-depth ablation studies on ViDROP design choices with a ViT-B. We use 16384 clusters for the DINO loss, both for the CLS token and the per-patch loss, and mask uniformly between 10% to 40% of the tokens. To accelerate training, we start from a pretrained VideoMAE [244] checkpoint trained on K400 for 800 epochs.

Table 6.1: **ViDROP ablation experiments.** We report linear probing (single crop) accuracy (%) with ViT-B/16 on K400. If not specified, the default is: the loss is iBOT, the data augmentation is random resized cropping, the number of small crops is 8, the masking ratio is 85%, and the pre-training length is 60 epochs. Default settings are marked in gray.

(a) **Loss function.** Patch loss with MASK tokens improves the performance.

loss	linear
stare	51.7
DINO	53.1
iBOT	**54.9**

(b) **Number of small crops.** Even having a few small crops boosts the performance.

num.	linear
8	**54.9**
4	54.1
0	50.2

(c) **Drop pattern.** Simple random token dropping is more effective than complex patterns.

pattern	linear
random tokens	**54.9**
random tubes	54.5
block (vjepa)	53.1

(d) **Drop rate**. Lower drop rates improve performance but require longer training times. The 80% drop rate model trained longest due to smaller batches.

rate	linear	time(hh:mm)
80%	55.2	31:54
85%	**54.9**	24:21
90%	54.4	23:46
95%	51.5	23:19

(e) **Number of clusters**. Reducing clusters from 65k to 16k maintains accuracy, while 2k clusters slightly decrease it. Using a shared head for 65k clusters also lowers accuracy.

num.	shared	linear
65k	x	**54.9**
16k	x	**54.9**
2k	x	54.3
65k	✓	53.8

(f) **Masking probability**. Increasing the masking probability range from 10-40% to higher values slightly improves model accuracy (all the models use 16k clusters).

min	max	linear
0.1	0.4	54.9
0.1	0.7	54.9
0.5	0.7	**55.5**
0.7	0.7	55.0

Loss function. Having a loss on patch tokens is one of the key components of ViDROP. We conduct experiments with different losses, as shown in Table 6.1a. Having an extra loss on tokens (iBOT) is better than not having it (DINO), and having the MASK tokens and calculating the loss on them is better than not having them and calculating the loss on all visible tokens (stare).

Number of small crops. A key success element of DINO [29] and MSN [7] was using small crops. We see a similar pattern in Table 6.1b, where small crops significantly boost performance. We use the same setting as DINOv2 [196], but observe that fewer crops are viable. If computational load is an issue (as small crops are not sparsified), the number can be reduced while maintaining considerable performance gains.

Drop pattern. Most previous video reconstruction methods have observed the importance of the token dropping pattern [18, 84, 244]. Surprisingly, in Table 6.1c, we show that simply randomly dropping the patches is more effective than tube dropping [244, 256] or block masking [18]. We notice that the self-supervised loss in the case of these patterns is higher than in the random case, and we suspect that the difficulty of block or tube masking is hindering the representational power of the model.

Table 6.2: **Effect of initialization on model performance.** The random model was trained for 240 epochs (4× the epochs of the pretrained models). Initial pretraining was conducted on 64 Tesla V100 GPUs, and our training on 4 RTX 4090 GPUs (approximately 2× the performance of V100). The VMAE_{1600} checkpoint initially achieves 43.5% accuracy, and ViT-Large checkpoint achieves 52.5%.

initialization	pretraining time	linear accuracy
random	N/A	53.3%
VMAE_{800}	27.7 hours	55.5%
VMAE_{1600}	55.4 hours	57.0%

Drop rate. Token dropping is an essential component for reducing the computational load of our model. In Table 6.1d, we can see that there is a trade-off in terms of training time and quality. Reducing the drop rate improves the quality but at the cost of extra training time. Note that in this experiment, we had to reduce the batch size of the model trained with an 80% token drop rate to be able to train it on our hardware. We further studied this trade-off and trained a dense model (i.e., a model with a drop rate of 0%) for 100 hours (almost $4\times$ the base model) and achieved an accuracy of only 47.1%, which is significantly lower than the 54.9% accuracy of the base model (see Figure 6.2).

Number of clusters. Since we are using Sinkhorn-Knopp centering for our DINO [29] losses, we are significantly more compute-heavy in the loss (compared to V-JEPA [18] and VideoMAE [244]). Additionally, we can't apply gradient accumulation with many steps, since the centering operation depends on the whole batch. Changing the loss is beyond the scope of this study, so instead, we reduced the number of clusters both for the global loss and patch loss. Results in Table 6.1e show that, similar to the findings of MSN [7], we can significantly reduce the number of clusters and maintain the same performance. We also notice that, similar to the findings of DINOv2 [196], having different heads for the two loss terms is beneficial. For the rest of the ablation studies, we used 16k clusters.

Masking probability. Following the setting of iBOT [299] and DINOv2 [196], we apply masking to only half of the large crops. For the other half, we initially randomly masked between 10% to 40% of the remaining tokens (after token dropping), slightly less than iBOT and DINOv2 that use 10% to 50%. Table 6.1f shows that we can use larger probabilities and achieve slight improvement. This is likely due to the heavy redundancies in video data compared to images (for example, it is common to use 75% sparsity for images in MAE [110] but 90% for videos [244]).

Initialization. To demonstrate the general applicability of our model, we trained it from scratch and compared it to models initialized with different pretrained weights, as shown in Table 6.2. The randomly initialized model was trained for 4× the epochs of the pretrained models but

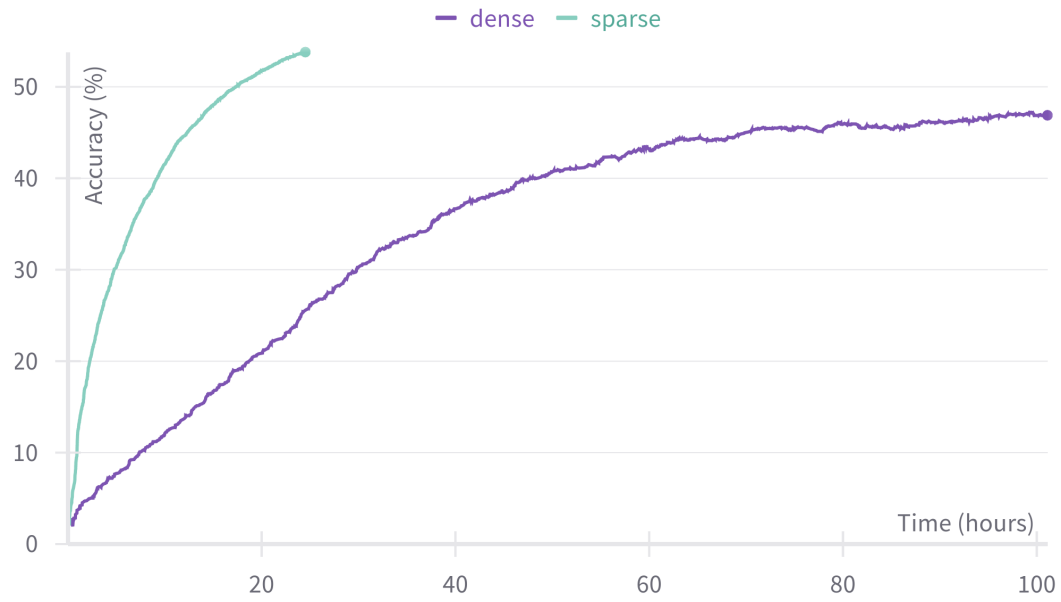


Figure 6.2: **Probing accuracy as a function of training time for dense and sparse models.** The dense model (without token dropping) takes significantly longer to train and achieves lower accuracy compared to the sparse model (with 85% token drop rate), even after training for four times longer.

still required significantly less total training time when considering the pretraining duration of the checkpoints.

Data compression method. To accelerate training for ViT-Small and ViT-Base models, we employed KMeans-based data compression. Table 6.3 shows that KMeans compression strikes a good balance between quality (measured by PSNR), encoding/decoding time, and compression rate. Other methods rely on large supervised models, complicating inference and disallowing the use of pretrained checkpoints. For all ablation experiments, data was compressed once (using Faiss [68]), taking around 40 hours for 60 epochs of data. We used a patch size of 10×10 and 65536 clusters.

Table 6.4 demonstrates that using KMeans data results in a $5.37\times$ speedup with a small performance cost when training a ViT-Small model. Training the same model on pixel data for the same duration yields worse performance. ViT-Base sees a $2.71\times$ speedup, and ViT-Large a $1.24\times$ speedup (better GPUs can lead to greater speedup, as data becomes the bottleneck).

Training throughput. In Table 6.5, we compare the training speed of various self-supervised learning (SSL) methods based on ViT for videos, using a single NVIDIA RTX 4090 GPU (with 24 GB VRAM). We employed the official code and configurations for each model: VJEPA [18] with repeated masking of 2, VideoMAEv2 [256] with 4, and VideoMAE [244] without repeated masking. DINOv2 represents the dense version of our model without sparsity.

Table 6.3: **Comparison of different compression methods.** Evaluation of various methods for compressing and decompressing 0.5M frames on 4 GPUs. KMeans-based methods offer a good balance between quality (PSNR), processing time, and compression factor.

Method	PSNR	Time (mm:ss)	Compression Factor
SD-XL	30.29	40:29	24
TinyAE-XL$_{\text{byte}}$	26.44	7:25	48
VQGAN-f16	23.36	25:59	384
KMeans$_{16\times16}$	24.28	1:23	384
KMeans$_{10\times10}$	25.75	1:41	150

Table 6.4: **Effect of KMeans on training speed and accuracy with ViT-Small.** Using KMeans compression achieves a $5.37\times$ speedup with a minor performance cost. Training on pixel data for the same duration results in lower performance.

KMeans	Time (hh:mm)	Linear Acc. (%)
✓	9:42	44.4
x	52:09	46.1
x	9:42	29.5

The measurements exclude data loading time and focus only on forward and backward times. We evaluated two architecture scales: ViT-Base and ViT-Large. Despite ViDROP having lower throughput due to the combination of small and large crops and heavier loss calculation, it is significantly more data efficient than reconstruction based methods, as shown in Table 6.2.

6.3.3 Results

For our main results, we train three different ViT-Large models to comprehensively evaluate the performance of ViDROP under various conditions. The first model, ViDROP$_{\text{vjepa}}$, is based on VJEPA [18] and employs only random resized cropping as data augmentation, representing a minimal augmentation approach. The second model, ViDROP$^{\text{aug}}_{\text{vjepa}}$, incorporates the heavy data augmentations used in DINO [29], allowing us to assess the impact of extensive augmentation techniques. Finally, ViDROP$^{\text{kmeans}}_{\text{vmae}}$ utilizes light data augmentations but is initialized with a VideoMAE checkpoint pretrained for 1600 epochs on Kinetics400 [139] and trained on KMeans compressed data. While KMeans compression doesn't offer significant speed benefits for ViT-Large models, we include this configuration to demonstrate the robustness of our findings. By using compressed data, we establish a lower bound on performance (as shown in Table 6.4), yet still outperform the original VideoMAE model. This underscores the effectiveness of ViDROP, even under potentially suboptimal data conditions. Through these three models, we showcase ViDROP's versatility and strong performance across various training scenarios and benchmarks.

Large action classification datasets. Table 6.6 presents our results on large action classification datasets, specifically Kinetics400 [139] (K400) and Something-Something-v2 [101] (SSv2). We focus primarily on linear probing [110] accuracies to evaluate the quality of SSL features

Table 6.5: **Training throughput of various SSL methods with ViT models.** Comparison of training speeds for different SSL methods using ViT-Base and ViT-Large architectures on a single NVIDIA RTX 4090 GPU. DINOv2 refers to the dense version of our model without sparsity. Our throughput is comparable to other reconstruction based models and significantly more than dense models (DINOv2 and SVT).

Method	ViT-Base		ViT-Large	
	Max BS.	TPUT	Max BS.	TPUT
SVT	4	8.4	1	1.5
VMAE	30	139.6	8	32.9
VMAEv2	15	58.8	8	29.7
VJEPA	37	74.4	22	37.9
DINOv2	12	23.4	4	7.8
ViDROP	33	43.1	11	17.4

without heavy fine-tuning or expensive heads. For completeness, we also include single-view attention pooling [18] results for ViT-based models.

For $\text{ViDROP}^{\text{kmeans}}_{\text{vmae}}$, we observe consistent improvements over the base VideoMAE model across all metrics. This demonstrates the effectiveness of our approach even when using compressed data. In the case of VJEPA-based models, ViDROP achieves state-of-the-art results in linear probing, significantly outperforming the base model and even surpassing the attention probing performance of the $\text{VJEPA}_{\text{Huge}}$ model on K400.

Interestingly, for our VJEPA-based models, attention probing yields lower performance than linear probing. This aligns with findings in the VJEPA paper [18], where a pretrained DINOv2 [196] model showed worse performance on ImageNet-1K [219] with attention probing compared to a linear head. On SSv2, our model shows lower performance compared to the base VJEPA checkpoint. This discrepancy can be attributed to the limited diversity in our pretraining data. While the VJEPA checkpoint leveraged VideoMix2M (which includes SSv2) for training, our pretraining utilized only K400 data, leading to a degree of forgetting for SSv2-specific features. Incorporating stronger data augmentations ($\text{ViDROP}^{\text{aug}}_{\text{vjepa}}$) narrows this gap significantly, but doesn't fully bridge the difference, highlighting the importance of diverse pretraining datasets.

Low-shot settings. To evaluate the effectiveness of our learned representations in scenarios with limited labeled data, we conducted experiments in low-shot settings on Kinetics-400. Table 6.7 presents the results of linear probing using varying percentages of the labeled training data. Our ViDROP models consistently outperform the baseline VJEPA across all data regimes, with particularly significant improvements in the most challenging low-shot scenarios. Notably, ViDROP achieves 58.5% accuracy with only 5% of the labeled data, surpassing VJEPA's

Table 6.6: **Performance comparison on large-scale action recognition datasets.** We report linear probing (LP) and attention pooling accuracies (%) on Kinetics-400 (K400) and Something-Something-v2 (SSv2). Numbers in parentheses indicate evaluation clips. ViDROP variants consistently outperform their baselines in LP, with $\text{ViDROP}^{\text{aug}}_{\text{vjepa}}$ achieving state-of-the-art performance. Note the performance drop on SSv2 for our models due to limited pretraining dataset diversity.

Method	K400		SSv2
	Linear (5×3)	Attention (1×1)	Linear (2×3)
ρBYOL [83]	71.5	-	25.3
SVT [212]	68.1	-	20.3
CVRL [206]	71.6	-	-
VideoMAE	52.5	68.6	27.9
$\text{ViDROP}^{\text{kmeans}}_{\text{vmae}}$	63.4	71.9	32.9
VJEPA	56.7	**73.7**	**43.2**
$\text{ViDROP}_{\text{vjepa}}$	72.4	71.1	33.8
$\text{ViDROP}^{\text{aug}}_{\text{vjepa}}$	**74.8**	72.7	38.7
$\text{VJEPA}_{\text{Huge}}$	-	74.0	-

performance (56.7%) when trained on the full dataset. This demonstrates the robustness and data efficiency of our learned representations. The augmented version, $\text{ViDROP}^{\text{aug}}_{\text{vjepa}}$, further improves upon these results, achieving 61.2% accuracy with just 5% of the data. These findings highlight the potential of ViDROP for real-world applications where labeled data may be scarce or expensive to obtain.

Small datasets. We evaluate our models on UCF-101 [233] and HMDB-51 [147] to assess representation transferability. Table 6.8 shows results for linear probing and K-nearest neighbors (KNN) classification. $\text{ViDROP}^{\text{kmeans}}_{\text{vmae}}$ significantly improves over VideoMAE, especially in KNN probing, with gains of 23.9 and 16.3 percentage points on UCF-101 and HMDB-51, respectively. For VJEPA-based models, we observe a trend similar to larger datasets. $\text{ViDROP}_{\text{vjepa}}$ shows slightly lower performance than the base VJEPA model, likely due to the data diversity issue mentioned earlier. However, $\text{ViDROP}^{\text{aug}}_{\text{vjepa}}$ with stronger data augmentations surpasses the baseline performance. It achieves state-of-the-art results in linear probing on both datasets and in KNN classification on HMDB-51, demonstrating our approach's effectiveness when combined with appropriate data augmentation strategies.

THUMOS14. To evaluate the generalization capability of our learned representations beyond action classification, we assess their performance on the temporal action detection task using

Table 6.7: **Low-shot learning performance on Kinetics-400.** Linear probing accuracies (%) are reported for different percentages of labeled training data. ViDROP variants consistently outperform the baseline VJEPA, with $\text{ViDROP}^{\text{aug}}_{\text{vjepa}}$ achieving the best results across all data regimes.

Method	5%	10%	50%	100%
VJEPA	43.6	48.8	52.0	56.7
$\text{ViDROP}_{\text{vjepa}}$	58.5	62.5	69.8	72.4
$\text{ViDROP}^{\text{aug}}_{\text{vjepa}}$	**61.2**	**65.4**	**72.5**	**74.8**

Table 6.8: **Performance comparison on small-scale action recognition datasets.** We report linear probing and KNN accuracies (%) on UCF-101 and HMDB-51. ViDROP variants show competitive performance, with $\text{ViDROP}^{\text{aug}}_{\text{vjepa}}$ achieving state-of-the-art results in most metrics.

Method	UCF-101		HMDB-51	
	Linear	KNN	Linear	KNN
ρBYOL	89.6	85.2	61.2	49.7
SVT	91.3	**87.2**	63.1	51.8
VideoMAE	84.5	49.1	60.3	29.2
$\text{ViDROP}^{\text{kmeans}}_{\text{vmae}}$	86.6	73.0	60.9	45.5
VJEPA	92.0	81.2	66.9	54.7
$\text{ViDROP}_{\text{vjepa}}$	91.1	81.3	66.0	51.0
$\text{ViDROP}^{\text{aug}}_{\text{vjepa}}$	**93.7**	84.8	**69.6**	**55.3**

the THUMOS14 [135] dataset. Table 6.9 presents the mean Average Precision (mAP) at different temporal Intersection over Union (tIoU) thresholds. Our ViDROP model demonstrates remarkable improvement over the VJEPA baseline across all tIoU thresholds. The average mAP of ViDROP (49.9%) more than doubles that of VJEPA (20.1%), indicating the superior quality and transferability of our learned representations for temporal action detection tasks. This substantial performance gain is particularly noteworthy given that both models use self-supervised features. For context, we also include results from ActionFormer [291]. While ActionFormer achieves higher performance, it's important to note that it utilizes supervised I3D features [30], which benefit from explicit supervision on action recognition tasks. In contrast, our ViDROP model uses purely self-supervised features, making its performance particularly impressive.

Table 6.9: **Temporal action detection results on THUMOS14.** We report mAP at different tIoU thresholds. ViDROP significantly outperforms VJEPA across all thresholds. *ActionFormer uses supervised I3D [30] features.

Method	0.3	0.4	0.5	0.6	0.7	Avg.
VJEPA	44.9	31.6	15.3	6.3	2.2	20.1
ViDROP$_{\text{vjepa}}$	64.4	59.1	52.2	42.7	31.3	49.9
ViDROP$^{\text{aug}}_{\text{vjepa}}$	70.2	65.5	59.1	49.5	36.6	56.2
ActionFormer*	82.1	77.8	71.0	59.4	43.9	66.8

6.4 Discussion

In this chapter, we introduced ViDROP, a novel self-supervised learning framework for video understanding that combines generative and discriminative learning paradigms within a single encoder-only architecture. By integrating token dropping and masking strategies, ViDROP addresses the critical challenge of prohibitive fine-tuning costs in large-scale video models while maintaining state-of-the-art performance across various benchmarks.

A key innovation of ViDROP is its k-means clustering-based video compression technique, which enhances efficiency while keeping representations close to the pixel space. This approach, coupled with the method's ability to perform well without heavy data augmentations, aligns with our thesis's focus on developing techniques that leverage existing knowledge while optimizing for computational efficiency.

The scalability of ViDROP across different network sizes and training regimes, along with its data efficiency in low-shot learning scenarios, highlights its potential for broad adoption in real-world applications. These characteristics demonstrate how strategic sparsity can lead to more adaptable and resource-efficient video analysis frameworks.

As we transition to the final chapter on SF-CLIP, we extend the principles of sparsity and efficient processing to multimodal learning, bridging visual and linguistic representations. This progression showcases the versatility of our approach across different modalities and represents a significant step in our exploration of efficient self-supervised visual representation learning in modern machine learning.

Chapter 7

SFCLIP: Building Vision-Language Models on Solid Foundations with Masked Distillation

Material in this chapter is based on: Sameni, Sepehr, Kushal Kafle, Hao Tan, and Simon Jenni. "Building Vision-Language Models on Solid Foundations with Masked Distillation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14216-14226. 2024.

As we conclude our exploration of sparsity and efficient processing in machine learning, this final chapter represents a significant leap forward in applying these principles beyond self-supervised learning (SSL). Building upon the insights gained from DILEMMA, SCALE, RIVER, and ViDROP, we now turn our attention to the challenging domain of Vision-Language Models (VLMs). This chapter introduces SF-CLIP (Solid Foundation CLIP), a novel approach that demonstrates how the concepts of strategic sparsity and efficient processing can be extended to weakly supervised models, particularly in the context of vision-language understanding.

The emergence of VLMs, exemplified by pioneering models like CLIP [209] and ALIGN [134], has been pivotal in integrating computer vision and natural language processing. These models have found applications in various domains, from text-guided image retrieval [26] to image captioning [158] and even text-to-image generation [138, 203, 205, 211]. However, despite their success, VLMs face limitations, including their reliance on noisy alt-text data, superficial understanding of visual content, and challenges in low-resource languages.

Recent works have explored various strategies to address these limitations, such as incorporating extra supervision [63, 186], using cleaner datasets [79], and re-captioning images [78, 190]. However, these approaches often compromise training efficiency or face scalability challenges, aligning with the broader theme of our thesis on balancing model capabilities with computational efficiency.

SF-CLIP addresses these challenges by leveraging the solid visual and linguistic understanding captured in foundational vision and language models. Our approach builds upon these pre-trained models through a combination of masked distillation and contrastive image-text pretraining. This strategy aligns with our thesis's focus on efficient processing and sparsity, as we sparsely apply distillation on a few examples at each step, maintaining higher training throughput while achieving better downstream performance.

Key to SF-CLIP's efficiency is its use of frozen text and image teacher models to provide per-token target latent representations during VLM training. This dense per-patch supervision enhances the spatial and compositional understanding of the image encoder, counteracting the tendency towards global feature learning in standard VLM training. This approach resonates with the sparsity concepts introduced in earlier chapters, as it allows for more efficient learning of fine-grained visual-textual relationships.

SF-CLIP also addresses the multilingual challenge in VLMs, a limitation that often requires extensive training on diverse languages [41, 290] or reliance on translation models [65]. By designing the VLM text input as a learned projection from fixed teacher word embeddings, SF-CLIP inherits multilingual capabilities despite being trained on monolingual data. This efficient transfer of linguistic knowledge aligns with our thesis's exploration of how strategic sparsity and efficient processing can unlock new capabilities in model training and application.

Our contributions can be summarized as follows:

- We introduce SF-CLIP, a novel approach to building VLMs that leverages foundational vision and language models through masked distillation.
- We demonstrate significant improvements in zero-shot and vision-language retrieval tasks while maintaining high training efficiency.
- Our method achieves multilingual proficiency and enhanced spatial understanding without requiring extensive multilingual training data or compromising computational efficiency.
- We show that selective application of distillation maintains higher training throughput while simultaneously achieving better downstream performance than prior methods using auxiliary training objectives.

By focusing on efficiency and leveraging pre-existing knowledge, SF-CLIP not only advances our understanding of VLMs but also aligns with the broader goals of this thesis in exploring how strategic sparsity and efficient processing can unlock new capabilities in model training and application. The subsequent sections will dive deeper into the methodology, experiments, and results, illustrating how SF-CLIP contributes to the evolving landscape of efficient and effective vision-language learning.

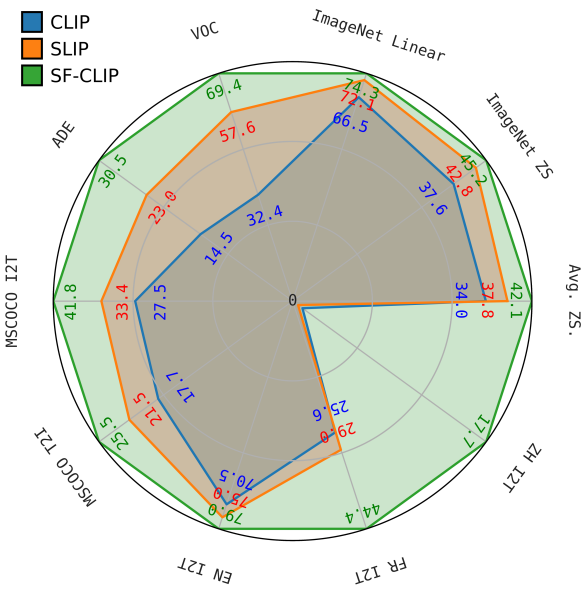


Figure 7.1: **SF-CLIP's Performance on Vision-Language Tasks.** Our model, SFCLIP, builds on the knowledge of foundational vision and language models to learn a joint embedding space. As a result, it not only shows improved zero-shot performance but also inherits strong multilingual and image segmentation capabilities from its teachers. The plot shows the performance of SFCLIP compared to SLIP and vanilla CLIP across ten established benchmarks. (all models are pretrained on YFCC-15M)

7.1 Background

This chapter extends the principles of multimodal weak supervised learning and Vision-Language Models (VLMs) discussed in Section 2.3. SF-CLIP addresses key challenges in current VLMs, particularly focusing on enhancing spatial-textual understanding while maintaining computational efficiency.

A crucial concept in SF-CLIP's approach is the combination of contrastive pretraining with masked knowledge distillation. This method builds upon the contrastive learning techniques outlined in Section 2.3.1, but aims to overcome the limitations in capturing fine-grained compositional details and complex visual-linguistic relationships, as highlighted in Section 2.3.2.

Another key aspect of SF-CLIP is its leveraging of foundational models, as discussed in Section 2.3.3. By incorporating these large-scale pretrained models, SF-CLIP aims to enhance the richness and depth of learned representations, addressing the challenges of spatial and linguistic understanding outlined in the main background chapter.

Lastly, SF-CLIP introduces a novel approach to multilinguality, not previously elaborated in Section 2.3.4. Instead of training directly on multilingual data or aligning with a pretrained multilingual encoder, SF-CLIP projects the multilingual embedding of a Large Language Model

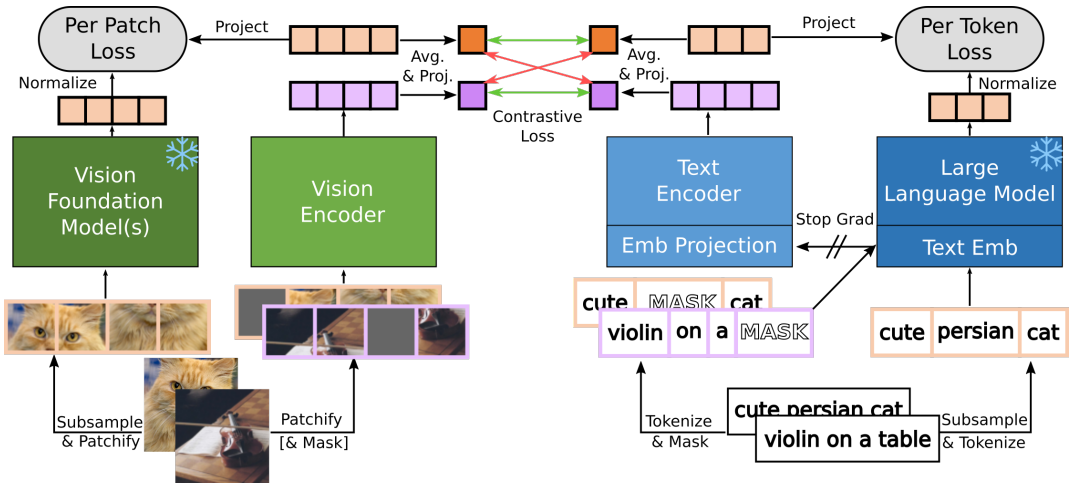


Figure 7.2: **Model Overview for SF-CLIP.** Our model learns to represent visual and textual data in a shared embedding space through an image and text encoder. We train our model on a dataset comprising image-text pairs, utilizing a combination of optionally masked feature distillation—aimed at inheriting the robust compositional understanding from vision and language foundation models—on a subset of the minibatch (illustrated by the orange sample in this figure) and standard vision-language contrastive learning to align the two modalities.

(LLM) to handle different languages. This method offers a potentially more efficient and flexible approach to multilingual VLM training, aligning with the thesis's focus on computational efficiency in self-supervised learning.

7.2 Method

We aim to learn a unified embedding space of text and images using two separate encoders, a visual encoder V and a text encoder T. We assume access to a paired dataset of images x_i and their noisy captions y_i (alt-text) $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$. Our model also leverages pre-trained teacher models V_{teacher} and T_{teacher} for both modalities. These teacher models are trained on potentially much larger uni-modal datasets (*e.g.*, large amounts of unlabelled images and texts), which are generally easier to obtain than the paired image-text data. All the models in our framework are based on the Transformer architecture [249], which encodes an input into a sequence of feature vectors, *e.g.*, $V(x) \in \mathbb{R}^{n_v \times d_v}$, where n_v is the number of tokens and d_v the latent feature dimension of the visual encoder. At a high level, our proposed training strategy combines the usual contrastive loss between paired data [209] with masked knowledge distillation objective to the teacher in each modality. We name our model SFCLIP and show an overview in Figure 7.2. The remainder of this section describes our model and training objective in detail.

7.2.1 Training Objectives

Vision-Language Contrastive Objective. We use a standard contrastive learning objective to align our model's vision and language embeddings. Concretely, let $v(x_i) \in \mathbb{R}^d$ be the embedding vector resulting from our visual encoder and $t(y_i) \in \mathbb{R}^d$ be the corresponding text embedding. In our implementation, we calculate $v(x_i)$ and $t(y_i)$ as the average of all the final layer token embeddings in the transformers V and T, followed by a learned linear projection in each modality (*e.g.*, projecting d_v to d for the visual encoder). Finally, the projected embeddings are l2-normalized. We follow prior works [134, 209] and use a symmetric InfoNCE loss [248] formulation

$$\mathcal{L}_{CLIP} = \mathcal{L}_{I \to T} + \mathcal{L}_{T \to I}, \tag{7.1}$$

with

$$\mathcal{L}_{I \to T} = -\frac{1}{B} \sum_{i}^{B} \log \frac{\exp(v(x_i) \cdot t(y_i)/\tau)}{\sum_{j=1}^{B} \exp(v(x_i) \cdot t(y_j)/\tau)}$$
$$\mathcal{L}_{T \to I} = -\frac{1}{B} \sum_{i}^{B} \log \frac{\exp(v(x_i) \cdot t(y_i)/\tau)}{\sum_{j=1}^{B} \exp(v(x_j) \cdot t(y_i)/\tau)},$$

where τ is a learned temperature parameter, and B is the size of a training mini-batch.

Masked Feature Distillation. We combine the above contrastive vision-text alignment objective with a feature distillation loss in both modalities. These distillation objectives aim to anchor the learned student representations with strong pre-trained visual and textual representations that capture well the structure of visual and textual data (solid foundations). Note that this implies that the student and teacher input tokenizer must result in the same number of tokens for any input. We, therefore, inherit the teacher language tokenizers in practice. Furthermore, we pose feature distillation in a masked setting, where the student only partially observes the input and must recover the latent teacher representation of masked and unmasked input tokens. This masked reconstruction task additionally steers the encoders to learn structural patterns in the inputs. Concretely, given teacher visual encoders V_{teacher} and text encoder T_{teacher}, and their corresponding student models V and T, we pose the distillation losses

$$\mathcal{L}_{VD} = \|V(M_v \odot x) - V_{\text{teacher}}(x)\|_2^2 \tag{7.2}$$

for the visual encoder and for the text encoder

$$\mathcal{L}_{TD} = \|T(M_t \odot y) - T_{\text{teacher}}(y)\|_2^2, \tag{7.3}$$

where M_v and M_t are masks that randomly zero out a set of student input tokens. Note that we layer normalize the outputs of both teacher models in the loss calculation and that we include a learned linear projection from the output of V and T to teacher output features (the teacher and student feature dimensions can be different).

Overall Training Objective. Our overall learning objective combines the CLIP loss with the distillation losses

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{CLIP} + \lambda_1 \mathcal{L}_{VD} + \lambda_2 \mathcal{L}_{TD}, \tag{7.4}$$

where λ_1 and λ_2 weigh the contribution of the distillation terms. In this multitask objective, we can interpret $\mathcal{L}_{CLIP}$ as aligning the two modalities while $\mathcal{L}_{VD}$ and $\mathcal{L}_{TD}$ anchor the visual and textual encoders with strong pre-existing representations of visual and textual data.

7.2.2 Efficient Training

Batch Subsampling for Efficient Training. Since our training includes distillation from large teacher networks (*e.g.*, LLMs for the text encoder), a naive implementation would result in much increased computational and memory demands and much lower training throughput. This is due to feeding every example in the mini-batch to the two teacher networks as well. To counteract this, we propose to perform the masked distillation objectives only on a small random subset of each training mini-batch. As our experiments show, this provides the positive influence of masked distillation while preserving high training throughput. Furthermore, it would be possible to pre-compute the teacher representations and avoid the online computation of targets during training, trading off additional storage requirements for virtually no training overhead compared to vanilla CLIP training.

Inheriting Teacher Word Embeddings. A large portion of the text encoder parameters are dedicated to learned word embeddings. Instead of learning these from scratch, we opt for a linear projection from the teacher's frozen word embeddings to T's hidden dimension. Besides accelerating training and enhancing downstream performance, we observe multi-lingual vision-language understanding capabilities emerging from this design when leveraging a multi-lingual text teacher. This occurs even without seeing any multilingual paired data during training of V and T.

7.3 Experiments

We conducted extensive experiments to validate our model design and to demonstrate its advantages over the conventional CLIP-style training approach for vision-language models. In these experiments, we employed a vision encoder based on the ViT-B/16 architecture [67] and CLIP's corresponding text encoder architecture [209] but with a non-causal attention (Similar to MaskCLIP [63] and CLIP🚀 [85]). Following the methodology of SLIP [186], we trained our model on YFCC15M (a subset of YFCC100M [241]) for 25 epochs with a batch size of 4096. Our default selection for the visual teachers included SAM-H/16 [142] and DINOv2-L/14 [196], and for the text teacher, we chose XGLM-1.7B [163] with word embedding projection. By default, we masked up to 25% of the text tokens and none of the vision tokens (evaluated in ablations) and employed a subset of 1024 images for the visual

Table 7.1: **Results on ImageNet and image-text retrieval.** Zero-shot and Linear probing accuracies on Imagenet (left) and zero-shot image-text retrieval on Flickr30K [202] and MS-COCO [162] datasets (right). Best results in **bold** and second best with underline.

		ImageNet		Flickr30K						MS-COCO					
		Zero	Linear	Image-to-text			Text-to-image			Image-to-text			Text-to-image		
	Method	Shot	Probe	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
YFCC-15M	CLIP [186]	37.6	66.5	52.9	79.6	87.2	32.8	60.8	71.2	27.5	53.5	65.0	17.7	38.8	50.5
	SLIP [186]	42.8	72.1	58.6	85.1	91.7	41.3	68.7	78.6	33.4	59.8	70.6	21.5	44.4	56.3
	MaskCLIP [63]	44.5	73.7	**70.1**	90.3	**95.3**	45.6	**73.4**	82.1	41.4	67.9	77.5	**25.5**	49.7	**61.3**
	$SLIP_{100ep}$ [85]	45.0	73.6	59.7	85.5	91.6	39.6	66.5	76.6	33.8	60.0	71.2	22.9	45.9	57.3
	SFCLIP	**45.2**	**74.3**	68.7	**90.4**	94.8	**46.2**	73.2	**82.7**	**41.8**	**68.3**	**78.4**	**25.5**	**50.2**	**61.3**
CC-12M	CLIP [78]	40.2	70.3	63.3	86.3	92.4	48.0	73.9	82.5	37.8	65.4	75.7	25.8	51.0	62.5
	SLIP [186]	40.7	73.7	62.5	87.2	92.1	46.6	73.3	80.9	37.6	64.9	75.5	26.8	51.4	62.7
	LaCLIP [78]	48.4	72.3	63.9	86.5	92.6	51.6	78.8	86.2	38.0	64.8	75.0	26.5	51.2	62.6
	LaSFCLIP	**53.6**	**75.3**	**71.8**	**91.9**	**95.2**	**59.9**	**84.2**	**90.9**	**44.3**	**71.3**	**80.2**	**31.4**	**57.3**	**68.1**

teachers and 512 for the text teacher. The values of λ_1 and λ_2 were consistently set to 1. To further demonstrate the broad applicability of our method, we also trained LaSF-CLIP, a language-augmented version of our SF-CLIP, using LaCLIP's language rewrites [78] on CC12M [32] for 35 epochs with a batch size of 8192, employing 1024 images for the vision teachers and 512 for the text teacher. For most of our comparisons, we used the official SLIP[1] and LaCLIP[2] checkpoints. All the models were trained using eight A100 GPUs on a single node using OpenCLIP's codebase [41].

7.3.1 Common Benchmarks

ImageNet Classification. We evaluate both zero-shot and linear probing accuracy of our model on ImageNet-1k [219]. We use the 7 prompts in SLIP [186] for zero-shot and 90 epochs of training with added batch normalization [127] without affine parameters [110] for the linear probing. As can be seen on the left of Table 7.1, SFCLIP performs better than all the other models with both pretraining datasets. Most notably, on CC-12M, SFCLIP gets a whopping 5.2% improvement over LaCLIP in the zeroshot setting.

Zero-shot Text/Image Retrieval. We report the zero-shot text-image retrieval results on two benchmark datasets, Flicr30K [202] and MS-COCO [162] on the right side of Table 7.1. Overall, in the case of YFCC-15M, we see on-par performance with MaskCLIP, and on CC-12M we see significant improvements over LaCLIP.

Zero-shot Classification on Small Datasets. We use the 20 datasets of the Image Classification in the Wild (ICinW) challenge [156] to assess our zero-shot classification accuracies in Table 7.2. Other than the datasets that most methods perform poorly on, like MNIST and

[1] https://github.com/facebookresearch/SLIP#vit-base

[2] https://github.com/LijieFan/LaCLIP#pre-trained-models

Aircraft (likely due to the domain gap between YFCC15M/CC12M and these datasets [63]), our method outperforms the others on average by 2% on YFCC15M and by 5% on CC-12M. Not only does our model outperform $\text{SLIP}_{\text{100ep}}$ (which was trained for 100 epochs, instead of 25), it even gets close to MaskCLIP [63] trained on significantly more data.

Table 7.2: **Zero-shot evaluation on ICinW classification benchmarks.** Best results in **bold** and second best with underline. Models are trained on YFCC-15M by default. Models marked with an asterisk * are trained on CC-12M. The model marked with a dagger † is trained on YFCC-15M+CC-3M+CC-12M+ImageNet-21K (ImageNet-1k is removed, around 13M images).

	CLIP [63]	*SLIP [63]*	*MaskCLIP [63]*	*CLIP 🚀 [85]*	*$\text{SLIP}_{\text{100ep}}$ [85]*	*CLIP 🚀 $_{\text{32ep}}$ [85]*	*SFCLIP*	*CLIP* [78]*	*SLIP* [78]*	*LaCLIP* [78]*	*LaSLIP* [78]*	*LaSFCLIP **	*MaskCLIP† [63]*
Caltech-101	58.6	70.9	72.0	72.8	74.0	**75.4**	72.2	77.4	77.6	83.3	82.8	**84.6**	86.4
CIFAR-10	68.5	82.6	80.2	71.3	79.2	67.1	**85.0**	64.9	80.7	75.1	82.0	**86.7**	95.3
CIFAR-100	36.9	48.6	**57.5**	38.9	50.4	37.8	53.6	38.5	46.3	43.9	50.2	**57.3**	78.3
Country211	10.8	11.8	12.6	14.6	11.5	**15.6**	12.0	5.1	5.7	8.9	**9.2**	**9.2**	11.6
DTD	21.4	26.6	27.9	28.0	26.2	30.3	**35.2**	19.4	25.1	31.0	30.1	**42.2**	33.0
EuroSAT	30.5	19.8	**44.0**	12.6	20.8	23.2	43.7	20.1	25.8	27.3	20.4	**35.9**	57.7
FER-2013	16.9	18.1	20.3	–	**36.5**	–	30.6	30.8	–	26.7	–	**34.9**	18.8
Aircraft	5.1	5.6	6.1	9.9	8.4	**11.2**	11.0	2.4	2.3	5.6	4.4	**7.3**	8.0
Food-101	51.6	59.9	64.9	61.5	63.3	63.0	**65.0**	50.8	52.5	60.7	62.9	**65.1**	78.9
GTSRB	6.5	**12.6**	8.5	10.0	11.7	8.1	10.3	7.3	6.0	12.7	10.1	**18.4**	17.3
Memes	51.1	51.8	52.0	52.9	**55.1**	54.3	49.6	52.1	–	52.9	–	**53.0**	52.8
KittiDis	25.9	29.4	34.3	**44.2**	35.2	35.6	32.9	**36.3**	–	16.9	–	29.7	16.0
MNIST	5.0	9.8	4.9	9.4	**17.1**	9.8	11.6	10.1	–	19.2	–	**19.3**	7.3
Flowers	52.7	56.3	57.0	58.4	61.3	**62.8**	59.5	33.2	29.2	39.9	37.4	**43.7**	74.2
Pets	28.6	31.4	34.3	30.7	34.7	35.4	**38.1**	64.1	58.6	72.4	70.6	**76.3**	74.4
PatchCam	51.7	**55.3**	50.1	51.1	52.1	51.6	54.1	50.3	–	50.6	–	**54.8**	52.1
SST2	**52.5**	51.5	49.9	50.4	49.9	50.1	50.3	47.6	–	48.4	–	**50.3**	46.2
RESISC45	22.4	28.5	35.7	37.2	27.8	36.0	**39.7**	38.9	36.6	44.3	45.6	**49.1**	54.3
Cars	4.5	5.4	6.7	6.7	8.1	**8.2**	**8.2**	24.1	24.9	**36.3**	32.2	35.7	26.5
Voc2007	79.1	80.5	**82.1**	–	78.67	–	80.2	77.0	–	81.9	–	**84.1**	82.3
Average	34.0	37.8	40.1	–	40.1	–	**42.1**	37.5	–	41.9	–	**46.9**	48.9

Table 7.3: **Zero-shot semantic segmentation (mIoU%).** We upsample the output feature map to the image resolution and classify each pixel using models trained on YFCC-15M.

Method	Pascal-Context	ADE-20K
CLIP [63]	13.5	7.2
MaskCLIP [63]	17.2	10.2
SFCLIP	**25.9**	**11.6**

Table 7.4: **Zero-shot instance segmentation.** We use the frozen SAM decoder with ViT-B/16 on the COCO dataset, both models are evaluated with 1024×1024 images using ground truth bounding boxes.

Method	Training Data	mAP	mAR
SAM [142]	SA-1B	57.8	60.8
SFCLIP	YFCC15M	45.0	54.6

7.3.2 Dense Visual Understanding

Zero-shot Semantic Segmentation. Even though CLIP was trained on whole images, DenseCLIP [298] showed that one can still get per-patch classifications from CLIP. Since our model inherits additional spatial understanding through our visual distillation objective, we expect to see improved performance compared to vanilla CLIP on zero-shot semantic segmentation. Following DenseCLIP, we use the final attention keys of the vision encoder and project them into the joint embedding space, where we apply a per patch zero-shot classification on Pascal-Context [184] and ADE-20K [297]. Table 7.3 shows that SFCLIP performs much better than CLIP and MaskCLIP (which also has a per patch loss).

Zero-shot Instance Segmentation. Since SFCLIP was trained to mimic SAM's [142] final representations, we can use SAM's decoder on top of our model "out of the box" and get better than chance results. In Table 7.4, we use the ground-truth bounding boxes of MS-COCO to get instance segmentations. Note that SFCLIP was only trained to predict SAM's features on YFCC-15M and was not trained to process 1024×1024 images, but still manages to learn something useful and compatible with SAM's decoder.

Linear Probing for Semantic Segmentation. Following the previous experiment, we also conducted linear probing on our per-patch representations for the task of semantic segmentation on Pascal-VOC [76], Pascal-Context [184], ADE-20K [297], and COCO-Stuff [25] datasets. Results in Table 7.5 show that SFCLIP indeed has a richer spatial representation than CLIP and significantly outperforms it in this task. Most notably on Pascal-VOC, SFCLIP is almost performing as well as CLIP trained on 1B images [91].

7.3.3 Better Language Understanding

Compositional Understanding Benchmark. Because of the noisy training data and the coarse contrastive loss, VLMs mostly act like a bag of words [287] and lack a deeper compositional understanding of the images. Previous benchmarks to assess compositional understanding like Winoground [242], VL-CheckList [295], ARO [287], CREPE [176], and Cola [213],

Table 7.5: **Linear head probing semantic segmentaion.** We report mIoU% on semantic segmentation datasets with ViT-B/16.

Method	Pascal VOC	Pascal Context	ADE 20K	MS COCO
Pretraining on YFCC-15M				
CLIP [186]	32.4	29.6	14.5	22.6
SLIP [186]	57.6	41.8	23.0	32.1
SFCLIP	**69.4**	**47.9**	**30.5**	**35.1**
Pretraining on CC-12M				
CLIP [78]	35.2	30.1	18.0	24.4
LaCLIP [78]	33.7	30.1	17.5	24.5
LaSFCLIP	**69.1**	**47.0**	**31.6**	**34.9**
Larger Dataset Pretraining, 448×448 Evaluation				
$\text{SAM}_{\text{SA-1B}}$ [142]	46.6	–	26.6	–
$\text{CLIP}_{\text{DataComp-1B}}$ [91]	70.7	–	36.4	–

were found to have shortcomings and be gameable in many cases. The SugarCREPE [120] benchmark aims to address those shortcomings and provides a more reliable metric to measure compositional understanding of the VLMs. Results in Table 7.6 (left) show that SFCLIP gets a decent improvement over prior dual encoder joint embedding VLM approaches on YFCC-15M but gets worse performance than CLIP with language rewrites [78]. This result shows that even though naively sampling from an LLM for text data augmentation can be useful for many tasks (as was shown in other experiments), the LLM hallucinations might make the model worse in some aspects at the end and just using the hidden representations of an LLM, like as in SFCLIP is a more reliable way than sampling from an LLM without looking at the image.

Verb Understanding Benchmark. VLMs often fail at identifying image-text pairs that show a mismatch concerning subjects, verbs, and objects. The SVO [111] benchmark aims to quantify this problem and identified that VLMs perform worse on verb understanding, likely due to the noisy training data. We see a clear improvement using SFCLIP in this task (Table 7.6, right), but we note that verbs remain challenging even for SFCLIP compared to subject and object, which are mostly nouns.

Multilingual Capabilities. YFCC15M, a subset of YFCC100M [241], primarily features English captions. Therefore, training a standard CLIP model on this data will not permit image/text retrieval with other languages. However, thanks to using a large multi-lingual language model as our text teacher (XGLM-1.7B [163]) and by inheriting its word embedding through a learned projection, SFCLIP shows out-of-the-box multilingual capabilities even though it was never explicitly trained on non-English data. Table 7.7 demonstrates that SFCLIP performs drastically better than baselines on XTD10 [1] for all languages. In contrast, we

Table 7.6: **Benchmarks on the shortcomings of VLMs.** SugarCREPE [120] (compositional understanding), and SVO [111](verb understanding).

		SugarCREPE				SVO			
	Method	Replace	Swap	Add	Average	Subject	Verb	Object	All
YFCC-15M	CLIP [186]	73.3	59.4	74.0	68.9	79.3	70.5	87.8	75.4
	SLIP [186]	75.2	58.6	73.7	69.2	80.3	72.8	**89.5**	77.4
	SFCLIP	**77.3**	**61.6**	**74.8**	**71.2**	**81.0**	**74.7**	87.1	**78.2**
CC-12M	CLIP [78]	**77.5**	61.8	**73.5**	**70.9**	80.8	76.9	89.5	80.0
	LaCLIP [78]	75.1	60.6	71.2	69.0	85.6	80.7	91.8	83.8
	LaSFCLIP	76.7	**63.3**	72.0	70.7	**87.8**	**84.0**	**94.2**	**86.7**

Table 7.7: **Image to text retrieval results on XTD10.** Comparison of I2T.R@5 (T2I.R@5 follows the same patterns) performance across different languages on the XTD10 [1] benchmark. We see a huge performance drop even on languages close to English for the baselines but SFCLIP sees a less severe drop. For the languages further from the English, CLIP and SLIP become as bad as random guessing but SFCLIP still performs significantly better than chance.

	Method	EN	ES	FR	IT	DE	RU	ZH	TR	JP	PL	KO
YFCC-12M	CLIP [186]	70.5	23.3	25.6	23.4	21.4	1.1	0.9	3.6	0.7	6.6	0.7
	SLIP [186]	75.0	26.8	29.0	22.1	21.7	0.3	0.5	3.8	0.7	7.5	0.6
	SFCLIP	79.0	48.7	44.4	43.1	41.3	32.5	17.7	14.8	10.4	9.4	6.5
CC-12M	CLIP [78]	78.9	4.3	10.8	8.5	7.2	0.7	0.4	2.3	1.0	4.2	0.5
	LaCLIP [78]	80.1	8.4	16.1	12.9	14.0	1.0	1.6	3.5	0.4	7.1	0.8
	LaSFCLIP	84.0	34.2	38.1	33.2	33.5	40.3	47.3	13.9	27.5	9.1	12.1

observe CLIP's and SLIP's performance drop strongly even for languages similar to English and nearing random levels for distant languages. SFCLIP, on the other hand, performs significantly better than chance even in very distant languages like Japanese and Korean. We believe it to be remarkable that just by learning a projection of input tokens and matching the outputs of the LLM in one language (*i.e.*, English in our case), the model can generalize well to various languages. We observe the same behavior on the models trained on CC-12M but with different languages. Based on our initial investigations, language rewrites [78] sometimes output sentences in Russian and Chinese, explaining the performance difference in these languages. This also shows that a small set of sentences in other languages can greatly benefit multilingual capabilities through teacher distillation.

Table 7.8: **Importance of the different components.** We study the different components that matter for different evaluation metrics. For the different variants, we highlight the differences from the default SFCLIP setting.

	$\text{Teacher}_{\text{txt}}$	Mask_{txt}	Emb. Proj.	$\text{Teacher}_{\text{img}}$	Mask_{img}	Subset Ratio	IN-ZS	MSC-T2I	MSC-I2T	ES-I2T	RU-I2T	Context
1	✓	✓	✓	✓	✗	12.5%	33.9	18.9	33.3	39.1	34.2	25.4
2	✗	✓	N/A	✓	✗	12.5%	33.2	18.6	31.5	18.4	1.6	25.3
3	✓	✗	✓	✓	✗	12.5%	33.6	18.9	32.5	38.0	31.1	25.2
4	✓	✓	✗	✓	✗	12.5%	35.2	18.8	31.9	22.6	0.8	23.8
5	✓	✓	✓	✗	✗	12.5%	31.3	16.6	29.0	35.5	33.8	22.7
6	✓	✓	✓	✓	✓	12.5%	34.3	18.7	32.3	37.4	33.6	25.6
7	✓	✓	✓	✓	✗	6.25%	33.9	18.8	32.9	38.9	34.1	25.2
8	✓	✓	✓	✓	✗	25%	34.1	18.7	31.5	39.0	32.4	25.2

7.3.4 Ablations

We perform extensive ablation experiments to verify the various design choices in our model and training algorithm. All the models in this section are trained on YFCC-15M for 8 epochs with a batch size of 4096. Unless stated otherwise we use a small 564M version of XGLM [163] and only DINO-L/14 as our vision teacher. For all of the evaluations we measured ImageNet-ZeroShot accuracy, MSCOCO text-to-image and image-to-text retrieval performance, image-to-text top 5 accuracies on a close to English language (ES) and distant language (RU), and finally zero shot semantic segmentation accuracy on Pascal-Context to have a full picture and compare models on many aspects.

Importance of Different Components. In Table 7.8, we report different model variants as we add or remove components. First, we can see that removing the text teacher or not inheriting the word embedding removes the multilingual capabilities (rows 2 and 4). If we remove the projected word embedding, we see better performance for ImageNet-ZeroShot but worse performance on everything else which indicates trade-offs between different benchmarks. Second, removing the image teacher leads to a drop in performance on all metrics (row 5). Next, we see consistent benefits for masking with text (row 3), while image masking (row 6) provides mixed results with ViT-B. Although we observe improvements from image masking for larger ViT architectures in initial exploration, we disable it by default for ViT-B. Lastly, adjusting the distillation rate to either double or half (rows 7 and 8) reveals an optimal performance at the default rate and indications of overfitting at higher rates.

On the Choice of Teachers. Our model supports various teacher combinations in both modalities. Table 7.9 compares these combinations, utilizing XGLM models (564M and 1.7B parameters) for text and DINO-L/14 and SAM-H/16 for vision. Generally, we note improved performance with an increased number and size of the teachers.

Training Efficiency. We compare training speeds with and without the proposed batch subsampling (12.5%) for the distillation objective in Table 7.10. SFCLIP trains at a high speed, only slightly slower than standard CLIP, but significantly faster than SLIP(CLIP+SimCLR) [186]

Table 7.9: **Effect of different teachers.** We study the effects of different teachers of varying sizes on VLM performance.

Text	Vision	IN-ZS	MSC-T2I	MSC-I2T	ES-I2T	RU-I2T	Context
-	-	31.5	15.5	28.0	16.9	0.8	21.6
-	DINO	33.2	18.6	31.5	18.4	1.6	25.3
564M	-	31.3	16.6	29.0	35.5	33.8	22.7
564M	DINO	33.9	18.9	33.3	39.1	34.2	25.4
1.7B	DINO	34.6	19.5	33.4	41.7	34.9	25.7
1.7B	DINO+SAM	36.2	20.6	36.0	41.7	36.2	25.7

Table 7.10: **Training time.** We compare the training time of different methods with ViT-Base.

	CLIP	CLIP+SimCLR	MaskCLIP	SFCLIP $_{\text{Full Batch}}$	SFCLIP $_{\text{Subsampled}}$
Training Time	1.00×	2.67×	1.75×	2.28×	1.20×

and MaskCLIP [63]. It also shows improved performance in other experiments, indicating greater computational efficiency.

7.4 Discussion

In this final chapter, we introduced SF-CLIP, a novel Vision-Language Model (VLM) approach that extends the principles of sparsity and efficient processing to multimodal learning. Building upon the concepts explored in previous chapters, SF-CLIP demonstrates how strategic knowledge distillation can be applied to enhance both visual and linguistic capabilities in VLMs while maintaining computational efficiency.

SF-CLIP's integration of contrastive image-text pretraining with masked knowledge distillation from unimodal teachers effectively leverages the strengths of foundational vision and language models. This approach aligns with our thesis's focus on balancing model capabilities with computational constraints, achieving notable improvements in zero-shot classification accuracy and image-text retrieval performance while maintaining training efficiency.

A key innovation of SF-CLIP is its use of frozen text and image teacher models to provide per-token target latent representations during VLM training. This dense per-patch supervision enhances the spatial and compositional understanding of the image encoder, addressing limitations identified in standard VLM training. The method's ability to inherit multilingual capabilities despite being trained on monolingual data showcases the efficient transfer of linguistic knowledge, aligning with our exploration of how strategic sparsity and efficient processing can unlock new capabilities.

Our experiments demonstrate SF-CLIP's effectiveness across various tasks and its promising multilingual retrieval performance. This versatility underscores the potential of our approach in addressing a wide range of vision-language challenges efficiently.

SF-CLIP represents a significant step forward in applying the principles of sparsity and efficient processing beyond self-supervised learning to weakly supervised models. By leveraging the knowledge of foundational models through strategic distillation, we have shown how these principles can be extended to complex multimodal tasks, opening new avenues for efficient and effective learning across diverse domains of artificial intelligence.

Chapter 8

Conclusions

In this thesis, we explored innovative approaches to enhance the efficiency and effectiveness of self-supervised visual representation learning through the strategic application of sparsity. Our journey began with the introduction of DILEMMA in Chapter 3, where we demonstrated how token sparsity and shape-sensitive regularization could improve both computational efficiency and the quality of learned representations in contrastive learning. We then extended these principles to the domain of video analysis with SCALE in Chapter 4, showcasing how spatio-temporal crop aggregation could enable efficient and scalable video representation learning.

Our exploration of sparsity in generative tasks led to the development of RIVER in Chapter 5, where we applied sparse conditioning to the challenging problem of video prediction. This work not only demonstrated the versatility of our sparsity-based approach but also highlighted its potential in implicit world modeling. Building upon these insights, we introduced ViDROP in Chapter 6, a novel framework that combines generative and discriminative learning paradigms within a single encoder architecture, further pushing the boundaries of efficient video understanding.

Finally, in Chapter 7, we extended our sparsity principles to the realm of multimodal learning with SF-CLIP, showcasing how strategic knowledge distillation could enhance both visual and linguistic capabilities in Vision-Language Models while maintaining computational efficiency.

Throughout this journey, we consistently demonstrated that thoughtful application of sparsity could not only reduce computational demands but often improve the quality and generalizability of learned representations. Our work spans various modalities and learning paradigms, from contrastive learning and video understanding to generative models and multimodal learning, illustrating the broad applicability of our approach.

As we reflect on the contributions of this thesis, several promising directions for future research emerge:

Optimizing Sparsity Patterns. While we primarily used random sparsity, exploring sophisticated criteria for information dropping could yield improvements. Although our results in

DILEMMA suggest this might not benefit contrastive learning 3.13, findings from EVEREST [125] indicate potential in reconstruction-based methods, warranting further investigation.
Sparsity as Data Augmentation. Our results, particularly with ViDROP, suggest that sparsity alone might serve as an effective form of data augmentation. Further investigation into this possibility could lead to more generalizable SSL methods that rely less on domain-specific augmentations and human assumptions about the data [185].
End-to-End Training for Long Video Understanding. Leveraging sparse models like ViDROP, we could explore end-to-end training of the backbone network in SCALE, potentially unlocking better long video understanding features while remaining computationally feasible.
Hybrid Masking and Token Dropping in Multimodal Learning. Inspired by the success of ViDROP, combining token dropping with the masking strategy used in SF-CLIP could potentially yield even better training performance in multimodal learning tasks.
Refined Sparse Conditioning in Video Generation. Extending the principles of RIVER, future work could explore denoising subsets of frame tokens while conditioning on sparse tokens from multiple past frames, rather than conditioning on a single frame in the past. This approach could maintain computational efficiency while potentially improving the quality of generated videos.
Generative Models as Feature Extractors. While we didn't use RIVER as a feature extractor, the success of methods using image generation models as feature extractors [157] suggests potential in exploring video generation models for feature extraction.
Fine-tuning for Dense Inference. Given that most of our models are used in a dense setting during inference, it would be interesting to explore a short but potentially expensive fine-tuning stage to adapt our models to work with dense inputs. To mitigate the cost, techniques like LoRA [121] could be employed, similar to the multiple pretraining stages of current large language models with increasing context sizes.
Adaptive Sparsity Rates. Both DILEMMA and ViDROP highlighted the importance of varying sparsity ratios during training. Implementing a method to handle multiple sparsity rates within the same minibatch, similar to NaViT [55], could further enhance the flexibility and performance of our models.

As we conclude this thesis, it is clear that the strategic application of sparsity holds immense potential for advancing the field of self-supervised visual representation learning. By continuing to explore and refine these techniques, we can push the boundaries of what is possible in efficient, scalable, and effective machine learning across various modalities and tasks. The work presented here lays a solid foundation for future research that can bring us closer to artificial systems with visual perception and reasoning capabilities that match or even surpass those of humans.

Bibliography

[1] Pranav Aggarwal and Ajinkya Kale. Towards zero-shot cross-lingual image retrieval. *arXiv preprint arXiv:2012.05107*, 2020. 96, 97

[2] Adil Kaan Akan, Erkut Erdem, Aykut Erdem, and Fatma Guney. Slamp: Stochastic latent appearance and motion prediction. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14708–14717, 2021. 58

[3] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34, 2021. 5, 12, 20, 36

[4] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022. 54

[5] Sotiris Anagnostidis, Gregor Bachmann, Lorenzo Noci, and Thomas Hofmann. The curious case of benign memorization. *ArXiv*, abs/2210.14019, 2022. 48

[6] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6816–6826, 2021. URL https://api.semanticscholar.org/CorpusID:232417054. 14

[7] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael G. Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *European Conference on Computer Vision*, 2022. URL https://api.semanticscholar.org/CorpusID:248178208. 36, 41, 46, 75, 78, 79

[8] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael G. Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15619–15629, 2023. URL https://api.semanticscholar.org/CorpusID:255999752. 76

[9] Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zach Beaver, Jana von Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, Vivek Natarajan, and Mohammad Norouzi. Big self-supervised models advance medical image classification. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3458–3468, 2021. 35

[10] Mohammad Babaeizadeh, Chelsea Finn, D. Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. *ArXiv*, abs/1710.11252, 2018. 50, 66

[11] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021. 15, 49, 50, 60, 66

[12] Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli. Efficient self-supervised learning with contextualized target representations for vision, speech and language. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:254685875. 76

[13] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. *ArXiv*, abs/2202.03555, 2022. URL https://api.semanticscholar.org/CorpusID:246652264. 12, 77

[14] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan Yuille, Trevor Darrell, Jitendra Malik, and Alexei A. Efros. Sequential modeling enables scalable learning for large vision models. *ArXiv*, abs/2312.00785, 2023. URL https://api.semanticscholar.org/CorpusID:265552038. 12

[15] Fan Bao, Chongxuan Li, Yue Cao, and Jun Zhu. All are worth words: a vit backbone for score-based diffusion models. *arXiv preprint arXiv:2209.12152*, 2022. 57

[16] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 10, 11, 12, 21, 36, 77

[17] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei A. Efros. Visual prompting via image inpainting. *ArXiv*, abs/2209.00647, 2022. URL https://api.semanticscholar.org/CorpusID:251979350. 12

[18] Adrien Bardes, Quentin Garrido, Jean Ponce, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv:2404.08471*, 2024. 6, 14, 15, 73, 76, 77, 78, 79, 80, 81, 82

[19] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *ArXiv*, abs/2102.05095, 2021. URL https://api.semanticscholar.org/CorpusID:231861462. 14

[20] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel J. Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *ArXiv*, abs/2108.07258, 2021. URL https://api.semanticscholar.org/CorpusID:237091588. 7, 9

[21] Florian Bordes, Randall Balestriero, and Pascal Vincent. Towards democratizing joint-embedding self-supervised learning. *ArXiv*, abs/2303.01986, 2023. URL https://api.semanticscholar.org/CorpusID:257353289. 14

[22] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 27

[23] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *ArXiv*, abs/1809.11096, 2019. 56, 63

[24] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini

Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. URL https://api.semanticscholar.org/CorpusID:218971783. 9, 12, 15

[25] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2016. URL https://api.semanticscholar.org/CorpusID:4396518. 95

[26] Min Cao, Shiping Li, Juntao Li, Liqiang Nie, and Min Zhang. Image-text retrieval: A survey on recent research and development. *ArXiv*, abs/2203.14713, 2022. URL https://api.semanticscholar.org/CorpusID:247763152. 87

[27] Fredrik Carlsson, Philipp Eisen, Faton Rekathati, and Magnus Sahlgren. Cross-lingual and multilingual clip. In *International Conference on Language Resources and Evaluation*, 2022. URL https://api.semanticscholar.org/CorpusID:250163904. 17

[28] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *ArXiv*, abs/2006.09882, 2020. URL https://api.semanticscholar.org/CorpusID:219721240. 11

[29] Mathilde Caron, Hugo Touvron, Ishan Misra, Herv'e J'egou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021. URL https://api.semanticscholar.org/CorpusID:233444273. 20, 24, 26, 28, 34, 36, 40, 43, 45, 76, 78, 79, 81

[30] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017. 57, 84, 85

[31] Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7608–7617, 2019. 53

[32] Soravit Changpinyo, Piyush Kumar Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages

3557–3567, 2021. URL https://api.semanticscholar.org/CorpusID:231951742. 93

[33] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020. URL https://api.semanticscholar.org/CorpusID:211096730. 1, 2, 11, 12, 20, 23, 26, 33, 34, 40, 45

[34] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 1, 2

[35] Xi Chen, Xiao Wang, Soravit Changpinyo, A. J. Piergiovanni, Piotr Padlewski, Daniel M. Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V. Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali: A jointly-scaled multilingual language-image model. *ArXiv*, abs/2209.06794, 2022. URL https://api.semanticscholar.org/CorpusID:252222320. 17

[36] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022. 12

[37] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15745–15753, 2020. URL https://api.semanticscholar.org/CorpusID:227118869. 11

[38] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2, 3, 11

[39] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021. 11, 20, 21, 23, 24, 34

[40] Zhongzhi Chen, Guangyi Liu, Bo Zhang, Fulong Ye, Qinghong Yang, and Ledell Yu Wu. Altclip: Altering the language encoder in clip for extended language capabilities. *ArXiv*, abs/2211.06679, 2022. URL https://api.semanticscholar.org/CorpusID:253511222. 17

[41] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible

scaling laws for contrastive language-image learning. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2829, 2022. URL https://api.semanticscholar.org/CorpusID:254636568. 17, 88, 93

[42] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023. 47, 48

[43] Hyungjin Chung, Byeongsu Sim, and Jong-Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12403–12412, 2022. 52

[44] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 27

[45] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv: Computer Vision and Pattern Recognition*, 2019. 66

[46] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Pre-training transformers as energy-based cloze models. *arXiv preprint arXiv:2012.08561*, 2020. 21

[47] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020. 10, 20, 21, 31

[48] Adam Coates, A. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011. 27

[49] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024. 12

[50] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 12

[51] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. Tclr: Temporal contrastive learning for video representation. *Computer Vision and Image Understanding*, 219:103406, 2022. 13

[52] Ishan Rajendrakumar Dave, Simon Jenni, and Mubarak Shah. No more shortcuts: Realizing the potential of temporal self-supervision. *ArXiv*, abs/2312.13008, 2023. URL https://api.semanticscholar.org/CorpusID:266374984. 13

[53] Aram Davtyan, Sepehr Sameni, and Paolo Favaro. Efficient video prediction via sparsely conditioned flow matching. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23206–23217, 2022. URL https://api.semanticscholar.org/CorpusID:254044658. 7

[54] Decord. Decord. https://github.com/dmlc/decord, 2019. Accessed: 2024-06-20. 14

[55] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M. Alabdulmohsin, Avital Oliver, Piotr Padlewski, Alexey A. Gritsenko, Mario Luvci'c, and Neil Houlsby. Patch n' pack: Navit, a vision transformer for any aspect ratio and resolution. *ArXiv*, abs/2307.06304, 2023. URL https://api.semanticscholar.org/CorpusID:259837358. 102

[56] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 24

[57] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International conference on machine learning*, pages 1174–1183. PMLR, 2018. 50, 66

[58] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019. URL https://api.semanticscholar.org/CorpusID:52967399. 9, 12, 40, 46, 47

[59] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021. 15

[60] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Genie: Higher-order denoising diffusion solvers. *ArXiv*, abs/2210.05475, 2022. 52

[61] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 2, 10, 20, 21

[62] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Bootstrapped masked autoencoders for vision bert pretraining. *ArXiv*, abs/2207.07116, 2022. 36

[63] Xiaoyi Dong, Yinglin Zheng, Jianmin Bao, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Maskclip: Masked self-distillation advances contrastive language-image pretraining. *ArXiv*, abs/2208.12262, 2022. URL https://api.semanticscholar.org/CorpusID:251799827. 17, 87, 92, 93, 94, 95, 99

[64] Michael Dorkenwald, Fanyi Xiao, Biagio Brattoli, Joseph Tighe, and Davide Modolo. Scvrl: Shuffled contrastive video representation learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4131–4140, 2022. URL https://api.semanticscholar.org/CorpusID:249017621. 13

[65] Gabriel Oliveira dos Santos, Diego A. B. Moreira, Alef Iury Ferreira, Jhessica Silva, Luiz Pereira, Pedro Bueno, Thiago Sousa, Helena de Almeida Maia, N'adia Da Silva, Esther Colombini, Helio Pedrini, and Sandra Avila. Capivara: Cost-efficient approach for improving multilingual clip performance on low-resource languages. *ArXiv*, abs/2310.13683, 2023. URL https://api.semanticscholar.org/CorpusID:264405728. 17, 88

[66] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9): 1734–1747, 2015. 10

[67] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020. URL https://api.semanticscholar.org/CorpusID:225039882. 2, 9, 10, 11, 14, 20, 21, 23, 41, 57, 58, 92

[68] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024. 80

[69] Sivan Doveh, Assaf Arbelle, Sivan Harary, Rameswar Panda, Roei Herzig, Eli Schwartz, Donghyun Kim, Raja Giryes, Rogério Schmidt Feris, Shimon Ullman, and Leonid Karlinsky. Teaching structured vision&language concepts to vision&language models. *ArXiv*, abs/2211.11733, 2022. URL https://api.semanticscholar.org/CorpusID:253734406. 16

[70] Sivan Doveh, Assaf Arbelle, Sivan Harary, Roei Herzig, Donghyun Kim, Paola Cascante-Bonilla, Amit Alfassy, Rameswar Panda, Raja Giryes, Rogério Schmidt

Feris, Shimon Ullman, and Leonid Karlinsky. Dense and aligned captions (dac) promote compositional reasoning in vl models. *ArXiv*, abs/2305.19595, 2023. URL https://api.semanticscholar.org/CorpusID:258987899. 16

[71] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021. 11

[72] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In *CoRL*, 2017. 51, 56, 60, 63, 64, 66

[73] Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. Are large-scale datasets necessary for self-supervised pre-training? *arXiv preprint arXiv:2112.10740*, 2021. 12

[74] Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Angel Bautista, Alexander Toshev, Vaishaal Shankar, Joshua M. Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models. *ArXiv*, abs/2401.08541, 2024. URL https://api.semanticscholar.org/CorpusID:267028705. 11

[75] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12868–12878, 2020. URL https://api.semanticscholar.org/CorpusID:229297973. 15, 50, 54, 57, 77

[76] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 2009. 29, 95

[77] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. Pyslowfast. https://github.com/facebookresearch/slowfast, 2020. 42

[78] Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving clip training with language rewrites. *ArXiv*, abs/2305.20088, 2023. URL https://api.semanticscholar.org/CorpusID:258987272. 16, 87, 93, 94, 96, 97

[79] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. *ArXiv*, abs/2309.17425, 2023. URL https://api.semanticscholar.org/CorpusID:263310452. 87

[80] Yuxin Fang, Li Dong, Hangbo Bao, Xinggang Wang, and Furu Wei. Corrupted image modeling for self-supervised visual pre-training. *ArXiv*, abs/2202.03382, 2022. 10

[81] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 178–178, 2004. 27

[82] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6201–6210, 2019. 42

[83] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross B. Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3298–3308, 2021. URL https://api.semanticscholar.org/CorpusID:233444206. 13, 14, 36, 41, 42, 44, 46, 83

[84] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *ArXiv*, abs/2205.09113, 2022. URL https://api.semanticscholar.org/CorpusID:248863181. 6, 13, 15, 39, 78

[85] Enrico Fini, Pietro Astolfi, Adriana Romero-Soriano, Jakob Verbeek, and Michal Drozdzal. Improved baselines for vision-language pre-training. *ArXiv*, abs/2305.08675, 2023. URL https://api.semanticscholar.org/CorpusID:258685688. 92, 93, 94

[86] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017. 49, 61

[87] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *Advances in neural information processing systems*, 29, 2016. 49, 61, 66

[88] Evelyn Fix and Joseph L. Hodges. Discriminatory analysis - nonparametric discrimination: Consistency properties. *International Statistical Review*, 57:238, 1989. URL https://api.semanticscholar.org/CorpusID:120323383. 12

[89] Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. Stochastic latent residual video prediction. In *ICML*, 2020. 51, 58, 66

[90] Letian Fu, Long Lian, Renhao Wang, Baifeng Shi, Xudong Wang, Adam Yala, Trevor Darrell, Alexei A. Efros, and Ken Goldberg. Rethinking patch dependence for masked autoencoders. *ArXiv*, abs/2401.14391, 2024. URL https://api.semanticscholar.org/CorpusID:267212172. 14

[91] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander J. Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alexandros G. Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets. *ArXiv*, abs/2304.14108, 2023. URL https://api.semanticscholar.org/CorpusID:258352812. 95, 96

[92] Xiaojie Gao, Yueming Jin, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. Accurate grid keypoint learning for efficient video prediction. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5908–5915. IEEE, 2021. 58, 59

[93] Yunhao Ge, Yao Xiao, Zhi Xu, Xingrui Wang, and Laurent Itti. Contributions of shape, texture, and color in visual recognition. *arXiv preprint arXiv:2207.09510*, 2022. 29, 30

[94] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 5, 19, 21

[95] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 10, 21

[96] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13505–13515, 2021. 14

[97] Rohit Girdhar, Alaaeldin El-Nouby, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Omnimae: Single model masked pretraining on images and videos. *arXiv preprint arXiv:2206.08356*, 2022. 36, 39

[98] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 15

[99] Shreyank N Gowda, Marcus Rohrbach, and Laura Sevilla-Lara. Smart frame selection for action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1451–1459, 2021. 14

[100] Priya Goyal, Quentin Duval, Jeremy Reizenstein, Matthew Leavitt, Min Xu, Benjamin Lefaudeux, Mannat Singh, Vinicius Reis, Mathilde Caron, Piotr Bojanowski, Armand

Joulin, and Ishan Misra. Vissl. https://github.com/facebookresearch/vissl, 2021. 1, 2, 3

[101] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter N. Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5843–5851, 2017. URL https://api.semanticscholar.org/CorpusID:834612. 36, 41, 43, 77, 81

[102] Jean-Bastien Grill, Florian Strub, Florent Altch'e, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *ArXiv*, abs/2006.07733, 2020. URL https://api.semanticscholar.org/CorpusID:219687798. 2, 11, 21, 36, 40

[103] Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Maskvit: Masked visual pre-training for video prediction. *arXiv preprint arXiv:2206.11894*, 2022. 15, 60

[104] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. 6, 15

[105] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. *ArXiv*, abs/2203.08414, 2022. URL https://api.semanticscholar.org/CorpusID:247476291. 12

[106] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *Advances in Neural Information Processing Systems*, 33:5679–5690, 2020. 13

[107] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *arXiv preprint arXiv:2205.11495*, 2022. 15, 50, 57

[108] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 33

[109] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF*

conference on computer vision and pattern recognition, pages 9729–9738, 2020. 11, 19, 36

[110] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Doll'ar, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, 2021. URL https://api.semanticscholar.org/CorpusID:243985980. 2, 5, 11, 12, 13, 14, 20, 21, 24, 33, 34, 36, 40, 43, 46, 76, 77, 79, 81, 93

[111] Lisa Anne Hendricks and Aida Nematzadeh. Probing image-language transformers for verb understanding. *arXiv preprint arXiv:2106.09141*, 2021. 96, 97

[112] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv: Learning*, 2016. 58

[113] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *ArXiv*, abs/1808.06670, 2019. 41

[114] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 62

[115] Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020. URL https://api.semanticscholar.org/CorpusID:219955663. 50, 54

[116] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 15

[117] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022. 15, 60

[118] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001. 50

[119] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *arXiv preprint arXiv:2206.07696*, 2022. 15, 50, 51, 53, 60, 63, 66

[120] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *arXiv preprint arXiv:2306.14610*, 2023. 96, 97

[121] J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. URL https://api.semanticscholar.org/CorpusID:235458009. 102

[122] Kaiqin Hu, Jie Shao, Yuan Liu, Bhiksha Raj, Marios Savvides, and Zhiqiang Shen. Contrast and order representations for video self-supervised learning. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7919–7929, 2021. 43

[123] Deng Huang, Wenhao Wu, Weiwen Hu, Xu Liu, Dongliang He, Zhihua Wu, Xiangmiao Wu, Mingkui Tan, and Errui Ding. Ascnet: Self-supervised video representation learning with appearance-speed consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8096–8105, 2021. 35

[124] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023. 16

[125] Sun-Kyoo Hwang, Jaehong Yoon, Youngwan Lee, and Sung Ju Hwang. Everest: Efficient masked video autoencoder by removing redundant spatiotemporal tokens. *ArXiv*, abs/2211.10636, 2022. URL https://api.semanticscholar.org/CorpusID:259188150. 14, 102

[126] iNaturalist. iNaturalist 2019 competition dataset. https://github.com/visipedia/inat_comp/tree/master/2019, 2019. Accessed: 2023-03-04. 27

[127] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 24, 42, 43, 93

[128] Md. Mohaiminul Islam and Gedas Bertasius. Long movie clip classification with state-space video models. *ArXiv*, abs/2204.01692, 2022. 42, 43

[129] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. *Advances in neural information processing systems*, 33:19545–19560, 2020. 29

[130] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Andrew Brock, Evan Shelhamer, Olivier J. H'enaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver io: A general architecture for structured inputs & outputs.

ArXiv, abs/2107.14795, 2021. URL https://api.semanticscholar.org/CorpusID:236635379. 15

[131] S. Jenni, Givi Meishvili, and Paolo Favaro. Video representation learning by recognizing temporal transformations. In *European Conference on Computer Vision*, 2020. URL https://api.semanticscholar.org/CorpusID:220665754. 13

[132] Simon Jenni and Paolo Favaro. Self-supervised feature learning by learning to spot artifacts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2733–2742, 2018. 10, 21

[133] Simon Jenni and Hailin Jin. Time-equivariant contrastive video representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9970–9980, 2021. 13

[134] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, 2021. URL https://api.semanticscholar.org/CorpusID:231879586. 16, 87, 91

[135] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014. 77, 84

[136] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:4037–4058, 2019. URL https://api.semanticscholar.org/CorpusID:62841734. 1, 2, 4

[137] Alexia Jolicoeur-Martineau, Ke Li, Remi Piche-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *ArXiv*, abs/2105.14080, 2021. 52

[138] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10124–10134, 2023. URL https://api.semanticscholar.org/CorpusID:257427461. 87

[139] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Apostol Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *ArXiv*, abs/1705.06950, 2017. URL https://api.semanticscholar.org/CorpusID:27300853. 36, 41, 44, 46, 77, 81

[140] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 42

[141] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014. 15

[142] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. *ArXiv*, abs/2304.02643, 2023. URL https://api.semanticscholar.org/CorpusID:257952310. 17, 92, 95, 96

[143] Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *ArXiv*, abs/2106.00132, 2021. 52

[144] Jannik Kossen, Mark Collier, Basil Mustafa, Xiao Wang, Xiaohua Zhai, Lucas Beyer, Andreas Steiner, Jesse Berent, Rodolphe Jenatton, and Efi Kokiopoulou. Three towers: Flexible contrastive learning with pretrained image models. *ArXiv*, abs/2305.16999, 2023. URL https://api.semanticscholar.org/CorpusID:258947644. 17

[145] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 27

[146] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 27

[147] Hilde Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso A. Poggio, and Thomas Serre. Hmdb: A large video database for human motion recognition. *2011 International Conference on Computer Vision*, pages 2556–2563, 2011. URL https://api.semanticscholar.org/CorpusID:206769852. 36, 41, 45, 77, 83

[148] Manoj Kumar, Mohammad Babaeizadeh, D. Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A conditional flow-based model for stochastic video generation. *arXiv: Computer Vision and Pattern Recognition*, 2020. 66

[149] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Ccvs: Context-aware controllable video synthesis. *Advances in Neural Information Processing Systems*, 34: 14042–14055, 2021. 15, 60, 66

[150] Guillaume Leclerc, Andrew Ilyas, Logan Engstrom, Sung Min Park, Hadi Salman, and Aleksander Madry. Ffcv: Accelerating training by removing data bottlenecks. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition*

(CVPR), pages 12011–12020, 2023. URL https://api.semanticscholar.org/CorpusID:259224879. 14

[151] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018. 15, 50, 64, 66

[152] Min-Seob Lee, Song Park, Byeongho Heo, Dongyoon Han, and Hyunjung Shim. Seit++: Masked token modeling improves storage-efficient training. *ArXiv*, abs/2312.10105, 2023. URL https://api.semanticscholar.org/CorpusID:266348450. 15

[153] Wonkwang Lee, Whie Jung, Han Zhang, Ting Chen, Jing Yu Koh, Thomas E. Huang, Hyungsuk Yoon, Honglak Lee, and Seunghoon Hong. Revisiting hierarchical approach for persistent long-term video prediction. *ArXiv*, abs/2104.06697, 2021. 15

[154] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7327–7337, 2021. URL https://api.semanticscholar.org/CorpusID:231880022. 14

[155] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785*, 2021. 11

[156] Chunyuan Li, Haotian Liu, Liunian Harold Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Yong Jae Lee, Houdong Hu, Zicheng Liu, and Jianfeng Gao. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. *Neural Information Processing Systems*, 2022. 93

[157] Daiqing Li, Huan Ling, Amlan Kar, David Acuna, Seung Wook Kim, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Dreamteacher: Pretraining image backbones with deep generative models. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16652–16662, 2023. URL https://api.semanticscholar.org/CorpusID:259924740. 15, 102

[158] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *ArXiv*, abs/2301.12597, 2023. URL https://api.semanticscholar.org/CorpusID:256390509. 16, 87

[159] Xiaoni Li, Yu Zhou, Yifei Zhang, Aoting Zhang, Wei Wang, Ning Jiang, Haiying Wu, and Weiping Wang. Dense semantic contrast for self-supervised visual representation

learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1368–1376, 2021. 11

[160] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *ArXiv*, abs/2110.05208, 2021. URL https://api.semanticscholar.org/CorpusID:238582773. 17

[161] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, et al. Mst: Masked self-supervised transformer for visual representation. *Advances in Neural Information Processing Systems*, 34, 2021. 32

[162] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 93

[163] Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, Ves Stoyanov, and Xian Li. Few-shot learning with multilingual language models. *ArXiv*, abs/2112.10668, 2021. URL https://api.semanticscholar.org/CorpusID:260651613. 92, 96, 98

[164] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 50, 52, 53, 54, 55

[165] Fangyu Liu, Guy Edward Toh Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2022. URL https://api.semanticscholar.org/CorpusID:248496506. 16

[166] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *ArXiv*, abs/2304.08485, 2023. URL https://api.semanticscholar.org/CorpusID:258179774. 16

[167] Xingbin Liu, Jinghao Zhou, Tao Kong, Xianming Lin, and Rongrong Ji. Exploring target representations for masked autoencoders. *ArXiv*, abs/2209.03917, 2022. URL https://api.semanticscholar.org/CorpusID:252118863. 76, 77

[168] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 54

[169] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco Nadai. Efficient training of visual transformers with small datasets. *Advances in Neural Information Processing Systems*, 34, 2021. 11

[170] Yue Liu, Junqi Ma, Yufei Xie, Xuefeng Yang, Xingzhen Tao, Lin Peng, and Wei Gao. Contrastive predictive coding with transformer for video representation learning. *Neurocomputing*, 482:154–162, 2022. 14

[171] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 11

[172] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theory*, 28:129–136, 1982. URL https://api.semanticscholar.org/CorpusID:10833328. 74, 77

[173] Guillaume Lorre, Jaonary Rabarisoa, Astrid Orcesi, Samia Ainouz, and Stephane Canu. Temporal contrastive pretraining for video action recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 662–670, 2020. 14, 36

[174] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 42

[175] Pauline Luc, Aidan Clark, Sander Dieleman, Diego de Las Casas, Yotam Doron, Albin Cassirer, and Karen Simonyan. Transformation-based adversarial video prediction on large-scale data. *arXiv preprint arXiv:2003.04035*, 2020. 60, 66

[176] Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. Crepe: Can vision-language foundation models reason compositionally? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10910–10921, 2023. 95

[177] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *ArXiv*, abs/1306.5151, 2013. 27

[178] Marco Marchesi. Megapixel size image creation using generative adversarial networks. *ArXiv*, abs/1706.00082, 2017. 56, 63

[179] Kangfu Mei and Vishal M. Patel. Vidm: Video implicit diffusion models. *ArXiv*, abs/2212.00235, 2022. 62

[180] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 37

[181] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, 2016. URL https://api.semanticscholar.org/CorpusID:9348728. 13

[182] Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. Verbs in action: Improving verb understanding in video-language models. *ArXiv*, abs/2304.06708, 2023. URL https://api.semanticscholar.org/CorpusID:258108363. 16

[183] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12486, 2021. 13

[184] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 95

[185] Théo Moutakanni, Maxime Oquab, Marc Szafraniec, Maria Vakalopoulou, and Piotr Bojanowski. You don't need data-augmentation in self-supervised learning. *ArXiv*, abs/2406.09294, 2024. URL https://api.semanticscholar.org/CorpusID:270440405. 76, 102

[186] Norman Mu, Alexander Kirillov, David A. Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. *ArXiv*, abs/2112.12750, 2021. URL https://api.semanticscholar.org/CorpusID:245424883. 17, 87, 92, 93, 96, 97, 98

[187] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34, 2021. 21

[188] Charlie Nash, João Carreira, Jacob Walker, Iain Barr, Andrew Jaegle, Mateusz Malinowski, and Peter Battaglia. Transframer: Arbitrary frame prediction with generative models. *arXiv preprint arXiv:2203.09494*, 2022. 14

[189] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning.

In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf. 27

[190] Thao Nguyen, Samir Yitzhak Gadre, Gabriel Ilharco, Sewoong Oh, and Ludwig Schmidt. Improving multimodal datasets with image captioning. *arXiv preprint arXiv:2307.10350*, 2023. 87

[191] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008. 27

[192] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 2, 10, 20, 21

[193] Pedro O O Pinheiro, Amjad Almahairi, Ryan Benmalek, Florian Golemo, and Aaron C Courville. Unsupervised learning of dense visual representations. *Advances in Neural Information Processing Systems*, 33:4489–4500, 2020. 11

[194] Marc Oliu, Javier Selva, and Sergio Escalera. Folded recurrent neural networks for future video prediction. In *ECCV*, 2018. 53

[195] Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996. URL https://api.semanticscholar.org/CorpusID:4358477. 5

[196] Maxime Oquab, Timoth'ee Darcet, Théo Moutakanni, Huy Q. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Huijiao Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *ArXiv*, abs/2304.07193, 2023. URL https://api.semanticscholar.org/CorpusID:258170077. 9, 12, 17, 73, 75, 76, 77, 78, 79, 82, 92

[197] Letitia Parcalabescu, Michele Cafagna, Lilitta Muradjan, Anette Frank, Iacer Calixto, and Albert Gatt. VALSE: A task-independent benchmark for vision and language models centered on linguistic phenomena. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8253–8280, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.567. URL https://aclanthology.org/2022.acl-long.567. 16

[198] Song Park, Sanghyuk Chun, Byeongho Heo, Wonjae Kim, and Sangdoo Yun. Seit: Storage-efficient vision training with tokens using 1% of pixel storage. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17202–17213, 2023. URL https://api.semanticscholar.org/CorpusID:257631701. 77

[199] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 27

[200] Gaurav Parmar, Richard Zhang, and Junyan Zhu. On aliased resizing and surprising subtleties in gan evaluation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11400–11410, 2022. 60

[201] Mandela Patrick, Yuki M. Asano, Bernie Huang, Ishan Misra, Florian Metze, João F. Henriques, and Andrea Vedaldi. Space-time crop & attend: Improving cross-modal video representation learning. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10540–10552, 2021. 14

[202] Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, J. Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *International Journal of Computer Vision*, 123:74 – 93, 2015. URL https://api.semanticscholar.org/CorpusID:6941275. 93

[203] Dustin Podell, Zion English, Kyle Lacey, A. Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *ArXiv*, abs/2307.01952, 2023. URL https://api.semanticscholar.org/CorpusID:259341735. 87

[204] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 29

[205] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *ArXiv*, abs/2209.14988, 2022. URL https://api.semanticscholar.org/CorpusID:252596091. 87

[206] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, H. Wang, Serge J. Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6960–6970, 2020. URL https://api.semanticscholar.org/CorpusID:221090567. 83

[207] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 9

[208] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 9, 12

[209] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. URL https://api.semanticscholar.org/CorpusID:231591445. 2, 9, 12, 16, 38, 87, 90, 91, 92

[210] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. *arXiv preprint arXiv:2006.10704*, 2020. 15, 51, 60, 66

[211] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. URL https://api.semanticscholar.org/CorpusID:248097655. 15, 87

[212] Kanchana Ranasinghe, Muzammal Naseer, Salman Hameed Khan, Fahad Shahbaz Khan, and Michael S. Ryoo. Self-supervised video transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2864–2874, 2021. URL https://api.semanticscholar.org/CorpusID:244800737. 13, 14, 36, 41, 42, 44, 74, 83

[213] Arijit Ray, Filip Radenovic, Abhimanyu Dubey, Bryan A Plummer, Ranjay Krishna, and Kate Saenko. Cola: How to adapt vision-language models to compose objects localized with attributes? *arXiv preprint arXiv:2305.03689*, 2023. 95

[214] Adrià Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Patraucean, Florent Altch'e, Michael Valko, Jean-Bastien Grill, Aäron van den Oord, and Andrew Zisserman. Broaden your views for self-supervised video learning. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1235–1245, 2021. URL https://api.semanticscholar.org/CorpusID:232417490. 13, 14, 35, 36, 41, 44, 45

[215] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *ArXiv*, abs/2010.04592, 2021. 40

[216] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the*

IEEE/CVF conference on computer vision and pattern recognition, pages 10684–10695, 2022. 16, 50, 54

[217] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015. 15, 57

[218] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, 71(599-607):6, 1986. 11

[219] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211 – 252, 2014. URL https://api.semanticscholar.org/CorpusID:2930547. 82, 93

[220] Sepehr Sameni, S. Jenni, and Paolo Favaro. Representation learning by detecting incorrect location embeddings. In *AAAI Conference on Artificial Intelligence*, 2022. URL https://api.semanticscholar.org/CorpusID:254368133. 7

[221] Sepehr Sameni, S. Jenni, and Paolo Favaro. Spatio-temporal crop aggregation for video representation learning. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5641–5651, 2022. URL https://api.semanticscholar.org/CorpusID:254096149. 7

[222] Sepehr Sameni, Kushal Kafle, Hao Tan, and Simon Jenni. Building vision-language models on solid foundations with masked distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14216–14226, 2024. 7

[223] Madeline Chantry Schiappa, Michael Cogswell, Ajay Divakaran, and Yogesh Singh Rawat. Probing conceptual understanding of large visual-language models. *ArXiv*, abs/2304.03659, 2023. URL https://api.semanticscholar.org/CorpusID:258040947. 16

[224] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. *ArXiv*, abs/2210.08402, 2022. URL https://api.semanticscholar.org/CorpusID:252917726. 17

[225] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36. IEEE, 2004. 59, 63, 64, 66

[226] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green ai. *Communications of the ACM*, 63(12):54–63, 2020. 3, 4

[227] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *European Conference on Computer Vision*, pages 154–171. Springer, 2020. 14

[228] Younggyo Seo, Kimin Lee, Fangchen Liu, Stephen James, and P. Abbeel. Harp: Autoregressive latent video prediction with high-fidelity image generator. *ArXiv*, abs/2209.07143, 2022. 15

[229] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *ArXiv*, abs/2107.06383, 2021. URL https://api.semanticscholar.org/CorpusID:235829401. 16

[230] Zhan Shi, Xu Zhou, Xipeng Qiu, and Xiaodan Zhu. Improving image captioning with better use of captions. *arXiv preprint arXiv:2006.11807*, 2020. 17

[231] Ivan Skorokhodov, S. Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3616–3626, 2021. 62

[232] Yang Song, Jascha Narain Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2021. 15, 53

[233] Khurram Soomro, Amir Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv*, abs/1212.0402, 2012. URL https://api.semanticscholar.org/CorpusID:7197134. 36, 41, 44, 77, 83

[234] Chen Sun, Arsha Nagrani, Yonglong Tian, and Cordelia Schmid. Composable augmentation encoding for video representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8834–8844, 2021. 14

[235] Dídac Surís, Ruoshi Liu, and Carl Vondrick. Learning the predictability of the future. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12607–12617, 2021. 14

[236] Alex Tamkin, Vincent Liu, Rongfei Lu, Daniel Fein, Colin Schultz, and Noah Goodman. Dabs: A domain-agnostic benchmark for self-supervised learning. *arXiv preprint arXiv:2111.12062*, 2021. 10

[237] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. Vimpac: Video pre-training via masked token prediction and contrastive learning. *arXiv preprint arXiv:2106.11250*, 2021. 14, 36

[238] Chenxin Tao, Xizhou Zhu, Gao Huang, Yu Qiao, Xiaogang Wang, and Jifeng Dai. Siamese image modeling for self-supervised vision representation learning. *arXiv preprint arXiv:2206.01204*, 2022. 14

[239] Alexa R Tartaglini, Wai Keen Vong, and Brenden M Lake. A developmentally-inspired examination of shape versus texture bias in machines. *arXiv preprint arXiv:2202.08340*, 2022. 19, 21

[240] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. In *NeurIPS*, 2021. 15

[241] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl S. Ni, Douglas N. Poland, Damian Borth, and Li-Jia Li. The new data and new challenges in multimedia research. *ArXiv*, abs/1503.01817, 2015. URL https://api.semanticscholar.org/CorpusID:195345989. 92, 96

[242] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5228–5238, 2022. URL https://api.semanticscholar.org/CorpusID:248006414. 16, 95

[243] Nenad Tomasev, Ioana Bica, Brian McWilliams, Lars Buesing, Razvan Pascanu, Charles Blundell, and Jovana Mitrovic. Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet? *arXiv preprint arXiv:2201.05119*, 2022. 35

[244] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *ArXiv*, abs/2203.12602, 2022. URL https://api.semanticscholar.org/CorpusID:247619234. 2, 6, 13, 35, 36, 39, 41, 46, 47, 73, 76, 77, 78, 79, 80

[245] S. Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1526–1535, 2018. 66

[246] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *ArXiv*, abs/1812.01717, 2018. 51, 57, 58, 64, 66

[247] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *ArXiv*, abs/1711.00937, 2017. URL https://api.semanticscholar.org/CorpusID:20282961. 15, 54, 77

[248] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018. URL https://api.semanticscholar.org/CorpusID:49670925. 10, 11, 14, 16, 36, 37, 40, 91

[249] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 9, 15, 42, 90

[250] Manisha Verma, Sudhakar Kumawat, Yuta Nakashima, and Shanmuganathan Raman. Yoga-82: a new dataset for fine-grained classification of human poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1038–1039, 2020. 19, 27

[251] Ruben Villegas, Arkanath Pathak, Harini Kannan, D. Erhan, Quoc V. Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. *ArXiv*, abs/1911.01655, 2019. 53

[252] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation. *arXiv preprint arXiv:2205.09853*, 2022. 15, 50, 51, 53, 57, 58, 60, 66

[253] Haoxiang Wang, Pavan Kumar Anasosalu Vasu, Fartash Faghri, Raviteja Vemulapalli, Mehrdad Farajtabar, Sachin Mehta, Mohammad Rastegari, Oncel Tuzel, and Hadi Pouransari. Sam-clip: Merging vision foundation models towards semantic and spatial understanding. *ArXiv*, abs/2310.15308, 2023. URL https://api.semanticscholar.org/CorpusID:264439297. 17

[254] Jue Wang, Gedas Bertasius, Du Tran, and Lorenzo Torresani. Long-short temporal contrastive learning of video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14010–14020, 2022. 14

[255] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:2740–2755, 2019. 42

[256] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14549–14560, 2023. URL https://api.semanticscholar.org/CorpusID:257805127. 6, 9, 12, 13, 14, 78, 80

[257] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. Bevt: Bert pretraining of video transformers. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14713–14723, 2022. 36, 39

[258] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 42, 43

[259] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021. 11

[260] Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and S Yu Philip. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International Conference on Machine Learning*, pages 5123–5132. PMLR, 2018. 53

[261] Zhou Wang, Alan Conrad Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004. 58

[262] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022. 50, 52, 53, 62

[263] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. *arXiv preprint arXiv:2112.09133*, 2021. 46

[264] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. *arXiv preprint arXiv:1906.02634*, 2019. 53, 60, 66

[265] Nevan Wichers, Ruben Villegas, D. Erhan, and Honglak Lee. Hierarchical long-term video prediction without supervision. *ArXiv*, abs/1806.04768, 2018. 15

[266] Olivia Wiles, João F. M. Carreira, Iain Barr, Andrew Zisserman, and Mateusz Malinowski. Compressed vision for efficient video understanding. In *Asian Conference on Computer Vision*, 2022. URL https://api.semanticscholar.org/CorpusID:252735173. 15, 77

[267] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019. 14

[268] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13587–13597, 2022. 14, 35

[269] Chaoxia Wu and Philipp Krähenbühl. Towards long-form video understanding. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1884–1894, 2021. 36, 41, 42, 43

[270] Chenfei Wu, Jian Liang, Lei Ji, F. Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pre-training for neural visual world creation. In *ECCV*, 2022. 60, 66

[271] Yu Wu, Linchao Zhu, Xiaohan Wang, Yi Yang, and Fei Wu. Learning to anticipate egocentric actions by imagination. *IEEE Transactions on Image Processing*, 30:1143–1152, 2020. 14

[272] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 11, 25

[273] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020. 29, 30

[274] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. 28

[275] Tete Xiao, Colorado J Reed, Xiaolong Wang, Kurt Keutzer, and Trevor Darrell. Region similarity representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10539–10548, 2021. 11

[276] Annie Xie, Dylan Losey, Ryan Tolsma, Chelsea Finn, and Dorsa Sadigh. Learning latent representations to influence multi-agent interaction. In *Conference on robot learning*, pages 575–588. PMLR, 2021. 49, 61

[277] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021. 11

[278] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16684–16693, 2021. 11

[279] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: a simple framework for masked image modeling. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9643–9653, 2021. URL https://api.semanticscholar.org/CorpusID:244346275. 12, 77

[280] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3333–3343, 2022. 14

[281] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 15

[282] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *arXiv preprint arXiv:2203.09481*, 2022. 15, 50, 53, 66

[283] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. Clevrer: Collision events for video representation and reasoning. *ArXiv*, abs/1910.01442, 2020. 50, 61, 62, 63, 64, 66

[284] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 17

[285] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. *ArXiv*, abs/2202.10571, 2022. 62

[286] Liangzhe Yuan, Rui Qian, Yin Cui, Boqing Gong, Florian Schroff, Ming-Hsuan Yang, Hartwig Adam, and Ting Liu. Contextualized spatio-temporal contrastive learning with

self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13977–13986, 2022. 14

[287] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2022. 16, 95

[288] Shuangfei Zhai, Navdeep Jaitly, Jason Ramapuram, Dan Busbridge, Tatiana Likhomanenko, Joseph Y. Cheng, Walter A. Talbott, Chen Huang, Hanlin Goh, and Joshua M. Susskind. Position prediction as an effective pretraining strategy. In *ICML*, 2022. 10

[289] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18102–18112, 2021. URL https://api.semanticscholar.org/CorpusID:244117175. 17

[290] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *ArXiv*, abs/2303.15343, 2023. URL https://api.semanticscholar.org/CorpusID:257767223. 17, 88

[291] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *European Conference on Computer Vision*, volume 13664 of *LNCS*, pages 492–510, 2022. 77, 84

[292] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016. 2, 10

[293] Yujia Zhang, Lai-Man Po, Xuyuan Xu, Mengyang Liu, Yexin Wang, Weifeng Ou, Yuzhi Zhao, and Wing-Yin Yu. Contrastive spatio-temporal pretext learning for self-supervised video representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3380–3389, 2022. 14

[294] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Raymond Fu. Image super-resolution using very deep residual channel attention networks. In *European Conference on Computer Vision*, 2018. 57

[295] Tiancheng Zhao, Tianqi Zhang, Mingwei Zhu, Haozhan Shen, Kyusong Lee, Xiaopeng Lu, and Jianwei Yin. Vl-checklist: Evaluating pre-trained vision-language models with objects, attributes and relations. *ArXiv*, abs/2207.00221, 2022. URL https://api.semanticscholar.org/CorpusID:250244105. 16, 95

[296] Yue Zhao and Philipp Krahenbuhl. Training a large video model on a single machine in a day. *ArXiv*, abs/2309.16669, 2023. URL https://api.semanticscholar.org/CorpusID:263135660. 14

[297] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 28, 95

[298] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, 2021. URL https://api.semanticscholar.org/CorpusID:251105026. 95

[299] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Loddon Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *ArXiv*, abs/2111.07832, 2021. URL https://api.semanticscholar.org/CorpusID:244117494. 11, 12, 21, 28, 36, 45, 76, 77, 79

Erklärung

gemäss Art. 28 Abs. 2 RSL 05

Name/Vorname: Sepehr Sameni

Matrikelnummer: 20-131-686

Studiengang: Informatik

Bachelor ☐ Master ☐ Dissertation ☒

Titel der Arbeit: Efficient Self-Supervised Visual Representation Learning via Sparsity

Leiter der Arbeit: Prof. Dr. Paolo Favaro

Ich erkläre hiermit, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche gekennzeichnet. Mir ist bekannt, dass andernfalls der Senat gemäss Artikel 36 Absatz 1 Buchstabe o des Gesetzes vom 5. September 1996 über die Universität zum Entzug des auf Grund dieser Arbeit verliehenen Titels berechtigt ist.

Bern, August 15, 2024

.....
Sepehr Sameni