THE UNIVERSITY OF CHICAGO

ON THE SYMBIOSIS OF GENERATIVE MODELING
AND REPRESENTATION LEARNING

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY

XIAO ZHANG

CHICAGO, ILLINOIS
AUGUST 2025

Copyright © 2025 by Xiao Zhang

To my family

ACKNOWLEDGMENTS

Pursuing a Ph.D. has been a unique and life-changing experience for me—a long journey filled with unforgettable moments, both rewarding and challenging. It has been an adventure marked by moments of excitement, especially when months of effort and uncertainty finally led to ideas and methods coming together. At the same time, this path was marked by setbacks, failures, and periods of intense doubt. However, through these ups and downs, I developed resilience, persistence, and learned how to approach problems with a clear scientific mindset. These lessons have not only shaped my research but have also influenced the way I approach challenges in life.

I feel incredibly fortunate to have worked with so many talented and supportive researchers throughout this journey. First of all, I would like to thank my advisor, Michael Maire, sincerely. His passion for research and commitment to solving fundamental, yet challenging, problems have constantly inspired me. His guidance, encouragement, and patience helped me navigate obstacles, stay motivated, and develop as an independent researcher. I'm deeply grateful for the trust and support he has given me throughout my Ph.D.

I would also like to express my sincere gratitude to my committee members: Rebecca Willett, David Forsyth, Anand Bhattad, and Greg Shakhnarovich. Their valuable feedback, thoughtful discussions, and continuous support played an important role in shaping the direction of my research. I truly appreciate the time, care, and expertise they shared with me.

In addition, I have had the privilege of collaborating with a wonderful group of researchers: David Yunis, Kevin Wu, Matthew Walter, Ruoxi Jiang, Sudarshan Babu, Seemandhar Jain, Tewodros Ayalew, Will Gao, Xiaoyan Xing, and Yanhong Li. Their ideas, insights, and encouragement have been invaluable to my work. I am grateful for their contributions and have greatly enjoyed every opportunity to learn, collaborate, and grow alongside them.

I am also thankful to my lab mates and friends who made this journey not only productive but also memorable: Deqing Fu, Haochen Wang, Jiading Fang, Jingtian Ji, Jiahao Li, Joe Zhou, Luzhe Sun, Shengjie Lin, Tri Huynh, Vincent Tan, Xiaodan Du, Xin Yuan, and Zain Sarwar. Their support and friendship made even the toughest days brighter, and I am thankful for the memories we created together.

I want to thank my master's advisor, Jianbo Shi, who first introduced me to the fascinating world of computer vision. His passion for the field and dedication to tackling core computer vision problems deeply inspired me and shaped my research direction.

Finally, I would like to thank my family, whose love and support have been the foundation of everything I have accomplished. None of this would have been possible without their constant encouragement, and I am endlessly grateful for their presence throughout this journey.

TABLE OF CONTENTS

II REPRESENTATION LEARNING FOR GENERATIVE MODELS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Generative modeling and representation learning are core pillars of modern machine learning and computer vision. In recent years, the field has progressed from analyzing existing visual data to building generative models that can synthesize realistic and diverse visual content. These models offer not only powerful tools for content creation but also a unique perspective on visual understanding—by learning to reconstruct visual structures, they reveal how patterns can be captured, organized, and computationally represented. This thesis investigates the bidirectional relationship between generative modeling and representation learning through two complementary perspectives.

The first part of this thesis focuses on enhancing the representation learning capabilities of generative models. We begin by identifying a key limitation in standard architectural designs, specifically, how residual connections in generative models tend to favor high-rank features, which biases learning toward low-level textures rather than semantically meaningful abstractions. To address this, we introduce a decayed residual connection that penalizes the contribution of skip connections, effectively encouraging the model to learn compact, low-rank representations. This design significantly improves both representation quality and generative performance in masked autoencoders and diffusion-based models.

However, although diffusion models inherently learn useful representations, obtaining a compact and coherent low-dimensional embedding remains difficult due to the distributed nature of the representation across multiple noise levels and layers. Inspired by classical spectral methods, we propose an efficient distributed spectral clustering algorithm that aggregates features from various stages of the model to form a compact, semantically rich embedding.

We further extend our analysis to the generative adversarial network (GAN) framework. Observing that GAN discriminators often learn meaningful features, we introduce a novel representation-aware learning objective along with a capacity-preserving regularization technique. This approach enhances the quality of features learned by the discriminator, yielding improvements that make them useful for downstream semantic tasks.

The second part of this thesis examines how learned representations can be used to enhance the quality of generation. We develop a hierarchical generative model that operates in a cascade of semantic spaces, ranging from global structure to fine-grained details, extracted from a pretrained visual encoder. A set of diffusion models is trained to sequentially reconstruct these semantic features using denoising objectives. We demonstrate that a semantic-aware latent representation, such as a 256-dimensional vector from a CLIP encoder, achieves a significantly higher compression ratio than traditional VAE latents, preserving almost all visual information in a 256×256 image. This architecture not only improves sample quality but also accelerates training and outperforms larger models that use more data.

Finally, we explore how physics-informed representations can further enhance generation capabilities. By incorporating an autoencoder with a latent bottleneck designed to reflect physical properties—specifically, intrinsic reflectance and lighting—we enable the model to disentangle and manipulate scene properties. This allows for unsupervised generation of albedo maps and realistic image relighting.

CHAPTER 1

INTRODUCTION

1.1 Motivation and Overview

Understanding the visual world has long been a central pursuit in computer vision. A fundamental challenge in this endeavor is converting raw pixel data into structured, semantically meaningful representations. This process—often formulated as embedding sensor-level observations into a latent space where perceptually or structurally similar patterns are mapped close together—is essential for numerous downstream tasks, including classification, detection, segmentation, and interpretability.

Supervised learning has proven effective in learning such visual representations using labeled data [58] or paired textual descriptions [195]. However, its reliance on extensive human annotation limits its scalability. To overcome this bottleneck, self-supervised learning—particularly contrastive approaches [97, 40, 34]—has emerged as a compelling alternative. These methods exploit data augmentations to learn invariance without the need for labels. Nevertheless, they are constrained by assumptions such as augmentation invariance, which may not hold uniformly across domains or tasks [243].

Generative models provide a promising and complementary direction. Rather than learning invariances through discriminative objectives, generative models capture the underlying data distribution by synthesizing new samples. To produce realistic and high-fidelity images, these models must internally learn and organize latent representations that reflect the key structures and variations in the data. Importantly, generative models operate under fewer assumptions about the input distribution, offering greater scalability than contrastive methods.

Two major families of generative models have gained prominence: latent-variable models, such as VAEs [136] and GANs [85, 194], which explicitly map inputs to compact latent codes for sampling, and denoising autoencoders (DAEs) [249, 105], which learn to reconstruct clean inputs from noisy versions. Despite their flexibility and theoretical appeal, these models often underperform contrastive learning approaches in terms of representation quality.

This thesis investigates the root causes of this gap and proposes methods to enhance the representation learning capabilities of generative models. In parallel, it explores how incorporating structured, semantically meaningful representations within generative frameworks can improve generation quality and control.

The overarching objective of this thesis is to advance generative modeling through the lens of representation learning, with two primary goals: (1) to enable scalable, label-free feature learning, and (2) to improve the fidelity, diversity, and semantic consistency of generated visual content. These two capabilities—learning representations and generating data—are fundamentally intertwined. Robust generation relies on strong internal representations, while improved representations can, in turn, elevate generative performance.

This dual perspective offers a unified framework for understanding and improving generative models. The following chapters investigate this bidirectional relationship in detail.

1.2 Chapter Layout

The first part of the thesis (Chapters 2–4) focuses on improving representation learning within generative models:

- **Chapter 2** addresses a key architectural limitation in generative representation learning—namely, the design of residual connections. We propose decayed identity shortcuts, a simple yet effective modification that introduces no additional learnable parameters and is compatible with both masked autoencoders and diffusion models. This design leads to significant improvements in both representation and generation quality. This work [290] is co-led with equal contributions.

- **Chapter 3** extends the study to generative adversarial networks (GANs), which learn features with minimal inductive bias. This work [288] proposes a discriminator design that integrates structure-aware adversarial objectives and a Lipschitz-aware regularization scheme. Our approach yields discriminative features on par with state-of-the-art contrastive methods—even in the absence of data augmentation, where most generative models struggle.

- **Chapter 4** introduces an efficient compression objective to extract lower-dimensional features from diffusion model representations, which are otherwise distributed across noise levels and model layers. The method scales to large datasets, reveals coherent visual patterns, and enables the interpretability of the generative process. This work [291] is also co-led with equal contributions.

The second part of the thesis (Chapters 5–6) shifts focus toward improving generation quality using learned representations:

- **Chapter 5** proposes a hierarchical generative framework that sequentially generates semantic representations using a cascade of diffusion models. These hierarchical latents guide the generation process along structured semantic pathways, capturing fine-grained details while enhancing image fidelity and diversity. This work [292] is also co-led with equal contributions.

- **Chapter 6** presents the Latent Intrinsic Model [289], a generative system that learns to disentangle intrinsic and extrinsic factors of image formation via a physically-informed bottleneck. Without manual supervision, the model discovers albedo-like representations and supports zero-shot relighting of images, demonstrating strong disentanglement and controllability.

Part I

REPRESENTATION LEARNING FROM GENERATIVE MODELS

CHAPTER 2
RESIDUAL CONNECTIONS HARM GENERATIVE REPRESENTATION LEARNING

We show that introducing a weighting factor to reduce the influence of identity shortcuts in residual networks significantly enhances semantic feature learning in generative representation learning frameworks, such as masked autoencoders (MAEs) and diffusion models. Our modification notably improves feature quality, raising ImageNet-1K K-Nearest Neighbor accuracy from 27.4% to 63.9% and linear probing accuracy from 67.8% to 72.7% for MAEs with a ViT-B/16 backbone, while also enhancing generation quality in diffusion models. This significant gap suggests that, while residual connection structure serves an essential role in facilitating gradient propagation, it may have a harmful side effect of reducing capacity for abstract learning by virtue of injecting an echo of shallower representations into deeper layers. We ameliorate this downside via a fixed formula for monotonically decreasing the contribution of identity connections as layer depth increases. Our design promotes the gradual development of feature abstractions, without impacting network trainability. Analyzing the representations learned by our modified residual networks, we find correlation between low effective feature rank and downstream task performance.

2.1 Introduction

Residual networks (ResNets) [95] define a connection structure that has achieved near-universal adoption into modern architectures for deep learning. At the time of their development, supervised learning (*e.g.,* ImageNet [58] classification) was the driving force behind the evolution of convolutional neural network (CNN) architectures. Residual networks solved a key issue: CNNs constructed of more than approximately 20 convolutional layers in sequence became difficult to train, leading to shallower networks outperforming deeper ones, unless additional techniques, such as auxiliary outputs [235] or batch normalization [114], were employed. Both ResNets, and their predecessor, highway networks [231] provide elegant solutions to this trainability problem by endowing the network architecture with alternative shortcut pathways along which to propagate gradients. Highway networks present a more general formulation that modulates these shortcut connections with learned gating functions. However, given their sufficient empirical effectiveness, the simplicity of ResNet's identity shortcuts makes them a preferred technique.

While solving the gradient propagation issue, residual connections impose a specific functional form on the network; between residual connections, each layer (or block of layers) learns to produce an update slated to be added to its own input. This incremental functional form may influence the computational procedures learned by the network [86]. Alternatives to residual and highway networks exist that do not share this functional form, but implement other kinds of skip-connection scaffolding in order to assist gradient propagation [150, 110, 298]. Thus, shortcut pathways, rather than a specific form of skip connection, are the essential ingredient to enable the training of very deep networks. Nevertheless, nearly all modern large-scale models, including those based on the transformer architecture [248] incorporate the standard residual connection.

This design choice holds, even as deep learning has shifted into an era driven by self-supervised training. The shift to self-supervision brings to the forefront new learning paradigms, including those based on contrastive [266, 97, 40, 34, 87], generative [85, 130, 105, 226, 229, 204], and autoencoding [136, 98, 156] objectives. Many systems in the generative and autoencoding paradigms rely on "encoder-decoder" architectures, often styled after the original U-Net [205], which contains additional long-range shortcuts between corresponding layers in mirrored symmetry about a central bottleneck. With representation learning as a goal, one typically desires that the middle bottleneck layer produce a feature embedding reflecting abstract semantic properties. The interaction of skip-connection scaffolding for gradient propagation with encoder-decoder architectures, self-supervised training objectives, and bottleneck representations has not been carefully reconsidered. This is a worrisome oversight, especially since, even in the supervised setting with standard classification architectures, prior work suggests that unweighted identity shortcuts may be a suboptimal design decision [214, 74].

Intuitively, identity shortcuts may not be entirely appropriate for capturing high-level, semantic features as they directly inject low-level, high-frequency details of inputs into outputs, potentially compromising feature abstraction. We explore this issue within generative learning frameworks, including masked autoencoders (MAEs) [98] and diffusion models [105], leading paradigms for self-supervised image representation learning and generation. Our experiments demonstrate that identity shortcuts significantly harm semantic feature learning in comparison to an alternative we propose: gradually decay the weight of the identity shortcut over the depth of the network, thereby reducing information flow through it (Figure 2.2). With increasing layer depth, our approach facilitates a smooth transition from a residual to a feed-forward architecture, while maintaining sufficient connectivity to train the network effectively. Unlike prior work on learned gating [231] or reweighting [214] mechanisms for residual connections, our method is a forced decay scheme governed by a single hyperparameter.

A parallel motivation for our design stems from Huh et al. [113], who show that features from residual blocks have higher rank than those produced by comparative feed-forward blocks. The smooth transition between residual and feed-forward behavior induced by our decay scheme regularizes deeper features toward exhibiting low-rank characteristics. Section 2.6 experimentally explores the correlation between our decayed identity shortcuts and low-rank feature representations. Figure 2.1 previews the corresponding improvements to feature learning. Our contributions are:

- We introduce decayed identity shortcuts, a simple architectural mechanism which enhances feature abstraction in masked autoencoders and diffusion models.
- We identify a key correlation between our decayed identity shortcuts and low-rank inductive bias, empirically validating that our method improves classification accuracy and yields low-rank features.
- Our design within an MAE yields a substantial performance boost on ImageNet-1K [58]:

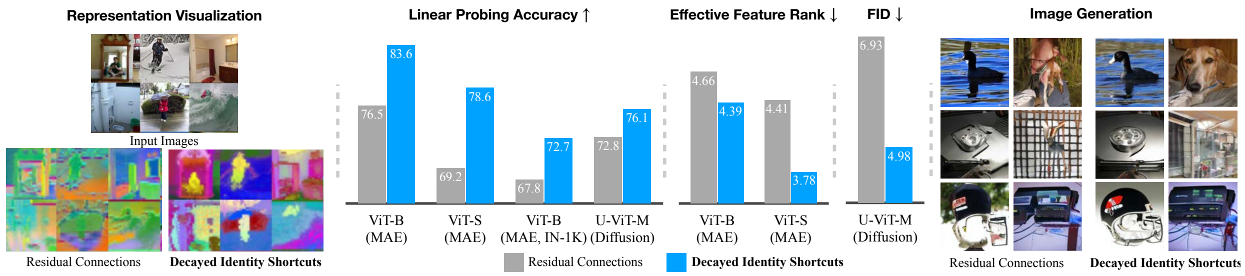


Figure 2.1: We design *decayed identity shortcuts* (Figure 2.2), a variant of residual connections, to facilitate self-supervised representation learning in generative model. Compared to standard residual connections, our approach yields superior abstract semantic features (*left*, visualized using Zhang et al. [291]'s approach), whose leading components pop out object instances and classes. Quantitative evaluation shows our architecture encourages lower feature rank and learns better feature representation for both MAE and diffusion models (*middle*), along with enhanced generation quality for diffusion models (*right*). These improvements require no additional learnable parameters.

achieving a linear probing accuracy of 72.7% (up from a baseline of 67.8%) and a K-Nearest Neighbor accuracy of 63.9% (an improvement from the baseline of 27.4%).

- In diffusion models, our design improves both feature learning and generation quality.
- Smaller models with decayed identity shortcuts outperform larger ones using standard residual connections.

2.2 Related Work

Self-supervised representation learning. Recent advancements [3, 137, 204, 240, 221, 199] in deep learning follow a common scaling law, in which a model's performance consistently improves with its capacity and the size of the training data. This effect can be observed in large language models (LLMs), which are trained on vast amounts of internet text, enabling them to perform some tasks at human level [151] and exhibit remarkable zero-shot capabilities [140]. These models are trained using next-token-prediction, allowing them to be trained without labeled data. In contrast, the progress of this scaling law in computer vision has largely depended on annotated data. For instance, the Segment Anything model [137] leverages 1 billion human-annotated masks, and state-of-the-art image generators [199] require training on huge datasets of text-image pairs [216]. However, the vast volume of unlabeled visual data and desire for continued scaling motivates a transition to self-supervised learning.

At present, two families of approaches to self-supervised visual representation learning appear particularly promising. **Contrastive representation learning** [266, 97, 40, 34, 87] achieves state-of-the-art performance in most downstream classification tasks by training discriminative models to maximize mutual information between differently augmented views of images. However, these approaches rely on extensive and intricate data augmentation pipelines, necessitating domain expertise for adaptation to newe domains. **Generative representation learning**, via masked image modeling [10, 98, 42], which trains to reconstruct occluded pixels, or via diffusion denoising [228, 105, 226], which trains to reverse a process that mixes images with Gaussian noise, relying less on forming discriminative augmentations, learns to extract representations inherently along the generative process. Some hybrid approaches [296, 111, 156] combine both families. Despite advancements, neither has demonstrated the same scalability [223] as seen in LLMs. This challenge is additional motivation for reconsidering the foundations of self-supervised network architectures.

Residual and skip-connection architectures. Highway networks [86] first propose an additive skip connection structure to provide a scaffolding for gradient propagation when training very deep (*e.g.,* 100 layer) networks. Motivated by the gating mechanisms within LSTMs [107], this solution uses learned gating functions to weight each combination of identity and layer output branches. Residual networks [95] are a simplification that removes these learned coefficients. DenseNet [110] and FractalNet [150] demonstrate that access to gradient paths of multiple lengths are the core requirement of training scaffolding, by introducing skip-connection structures with other functional forms. DenseNet utilizes feature concatenation instead of addition, while FractalNet imposes a recursive tree-like architecture combining subnetworks of multiple depths.

Zhu et al. [298] explore variants of ResNets and DenseNets with fewer points of combination between different internal paths, demonstrating that a sparser scaffolding structure may be more robust as network depth increases to thousands of layers. Savarese and Figueiredo [214] add a scalar gating functional to the layer output in residual networks, yielding a hybrid design between residual and highway networks; learning this scalar gating provides a consistent benefit to classification accuracy. Fischer et al. [74] develop a weighting scheme for residual connections based upon a sensitivity analysis of signal propagation within a ResNet. To date, none of these potential improvements have seen broad adoption.

Low rank bias in neural networks. Over-parameterized neural networks exhibit surprising generalization capabilities, a finding seemingly in contradiction with classical machine learning theory [184]. This phenomenon implies the existence of some form of implicit regularization that prevents the model from overfitting. From the perspective of neural network parameterizations, Arora et al. [6] suggest that linear models with more layers tend to converge to minimal norm solutions. In the context of CNNs, Huh et al. [113] demonstrate that stacking more feed-forward layers compels the model to seek lower rank solutions, and Jing et al. [120] reinforce this finding by adding more layers to an autoencoder's bottleneck, thereby creating a representation bottleneck. In vision transformers, Geshkovski et al. [80] examine the connection between attention blocks and mean-shift clustering [47], showing that repeated attention operations result in low-rank outputs. Moreover, Dong et al. [65] reveal that eliminating the shortcut connection from residual attention blocks causes features to degenerate to rank 1 structures doubly exponentially. From a different perspective, recent work [196, 15, 197] shows training algorithms implicitly induce low-rank behavior in neural networks. Radhakrishnan et al. [197] study the dimensionality reduction behavior of a recursive feature machine [196] and effectively verify performance on low-rank matrix recovery.

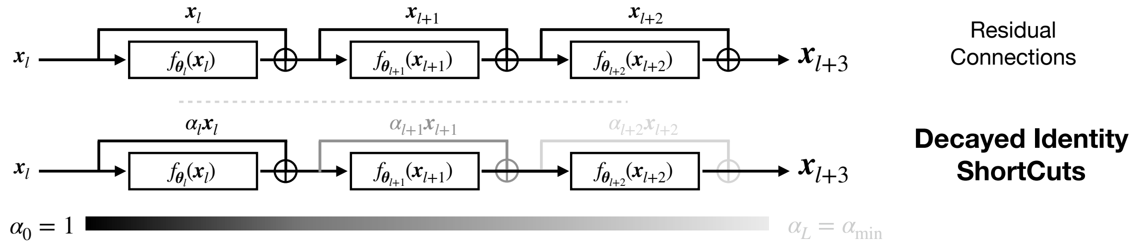


Figure 2.2: Our *decayed identity shortcuts* introduce a depth-dependent scaling factor to shortcuts in a residual network, thereby modulating the contribution of preceding layers and fostering greater abstraction in deeper layers. A simple schema for controlling decay factor α suffices to improve feature learning in both MAEs and diffusion models, as well as diffusion model generation quality.

2.3 Method

Prior works show that deeper feed-forward architectures have an inductive bias towards producing low-rank feature maps, while ResNets do not display the same behavior [113]. However, despite this bias, deeper feed-forward architectures are typically less effective and generalize worse than ResNets [95]. We aim to combine the properties of both feed-forward networks and ResNets, using the low-rank prior to enhance the abstraction capability of the network while maintaining the core benefits of the residual block, including stable training and the capacity to construct deeper models.

2.3.1 Decayed Identity Shortcuts

Feed-forward layers. Consider a neural network of L layers. For each layer l parameterized with θ_l, the operation of a feed-forward neural network can be described as:

$$\boldsymbol{x}_{l+1} = f_{\theta_l}(\boldsymbol{x}_l), \tag{2.1}$$

where $\boldsymbol{x}_l \in \mathbb{R}^d$ represents the output from the preceding layer, and f_{θ_l} denotes the network block applied at the current layer. Although it is widely known that pure feed-forward architectures are susceptible to vanishing gradients when building deeper models, Huh et al. [113] demonstrates that feed-forward modules offer implicit structural regularization, enabling deep models to generate abstract representations at bottlenecks.

Residual connections. To address the optimization problem of vanishing gradients in deeper neural networks, ResNets [95] construct each layer as a residual function, resulting in a modification to Eqn. 2.1:

$$\boldsymbol{x}_{l+1} = \boldsymbol{x}_l + f_{\theta_l}(\boldsymbol{x}_l). \tag{2.2}$$

This design builds shortcuts from input to output, allowing gradient magnitude to be preserved regardless of the depth of the model. However, a consequence of this design is that the output stays close to the input in practice [86], defeating the need to construct complex transformations over depth. The same phenomenon is also observed in highway networks [231], which adopt learnable gates $H_\phi(\boldsymbol{x}) \in [0, 1]^d$ in both the residual and skip branches: $\boldsymbol{x}_{l+1} = H_\phi(\boldsymbol{x}_l) \cdot \boldsymbol{x}_l + (1 - H_\phi(\boldsymbol{x}_l)) \cdot f_{\theta_l}(\boldsymbol{x}_l)$. Although this flexible design allows the model to build the abstraction level over depth, similar to feedforward networks, Srivastava et al. [232] finds $H_\phi \approx 1$ for most units, suggesting the model prefers copying the input.

Decayed identity shortcuts for unsupervised representation learning. Setting aside the optimization benefits brought by residual connections, we rethink the role of the residual connections from the viewpoint of representation learning. Abstraction can be viewed as invariance to local changes of input and is crucial to the disentanglement of the feature space [18]. Prior work suggests that a shortcut path of residual connections tends to preserve high-frequency fine-grained input information [86], resulting in decreased feature abstraction. We hypothesize that this lack of abstraction harms the capability of the model to learn meaningful low-level features and that ensuring an abstract structure in the deeper layers of the neural network will help improve representation learning, especially for unsupervised tasks that often use indirect proxy objectives, such as pixel-wise reconstruction loss. Motivated by this hypothesis, we propose to downweight the contribution from the shortcut path:

$$\boldsymbol{x}_{l+1} = \alpha_l \boldsymbol{x}_l + f_{\theta_l}(\boldsymbol{x}_l), \tag{2.3}$$

where $\alpha_l \in [0, 1]$ is a rescaling factor to the residual path, controlling the information flow through the skip connection. Fully expanding this relation for a network with L layers, we have that:

$$\boldsymbol{x}_L = \left(\prod_{l=0}^{L-1} \alpha_l\right) \boldsymbol{x}_0 + \sum_{l=0}^{L-2} \left(\prod_{i=l+1}^{L-1} \alpha_i\right) f_{\theta_l}(\boldsymbol{x}_l) + f_{\theta_{L-1}}(\boldsymbol{x}_{L-1}). \tag{2.4}$$

We see that the contribution of the input $\boldsymbol{x}_0$ is scaled by each $\alpha_l \leq 1$ while each subsequent network block output $f_{\theta_l}(\boldsymbol{x}_l)$ omits scaling factors up to α_l. Hence, the contribution of early features of the network is especially down-weighted, preventing the network from passing fine-grained detailed information to the bottleneck X_L. During our experiments, we find the effective decay factor of the final layer, $\alpha_L^{\text{eff}} = \prod_{l=0}^{L-1} \alpha_l$, plays a critical role in deciding the optimal decay rate when varying the network depth L.

Decay schema. Instead of specifying α_l as a constant across all layers, we choose α_l to be a function parameterized by the layer index l, where the contribution from the shortcut path is monotonically decreasing when l increases:

$$\alpha_l = 1 - \delta_\alpha l, \tag{2.5}$$

where $\delta_\alpha := \frac{(1-\alpha_{\min})}{L}$, $\alpha_L \equiv \alpha_{\min}$ is a minimum scaling factor applied at the final layer L. Our formulation brings two primary benefits. First, α_l, as a linear interpolation between 0 and 1, acts as a smooth transition between residual connections and feedforward layers, bringing us the optimization benefits seen in the residual connections, while simultaneously encouraging learning the deeper layers to learn more abstract representations. Second, similar to the naive formulation, our method only introduces one extra hyperparameter $\alpha_{\min}$, which is not data-dependent and does not need to be learned.

2.3.2 Implementation Strategy

Skip connections for autoencoders. Since our method progressively decays the residual connections over network depth, it encourages the most abstract features to be learned by later layers. However, learning an abstract bottleneck is detrimental to the training objectives that aim for pixel-wise reconstruction, as they necessitate the preservation of detailed information. To address this, we incorporate standard skip connections between the encoder and decoder, enabling the encoder to directly pass information from shallow layers to the decoder while learning increasingly abstract representations in the deeper encoder layers.

Stabilizing training with residual zero initialization. The model exhibits rapid feature norm growth at the beginning of training for $\alpha_{\text{min}} \leq 0.7$. We suspect that the model learns to amplify the output feature norm of $f_{\theta_l}(\boldsymbol{x})$ to counteract the significant decay applied to the residual connection. This growth leads to training instability and negatively impacts training convergence. To address this issue, we follow the implementation of previous works [105] and initialize the weights of the final output layer in each f_{θ_l} to zero instead of using the original Xavier uniform initialization [83]. This approach enhances training stability by controlling the growth of feature norm, especially with smaller α_{min}.

2.4 Experiments on Masked Autoencoder (MAE)

For masked autoencoders (MAEs) [98], we replace the residual connections in the encoder's MLP and attention blocks with decayed identity shortcuts. The MAE operates by accepting images with a random subset of pixels masked out and learning to recover the discarded pixels. Since the original MAE has twice the number of encoder layers as decoder layers, we build encoder-decoder skip connections by injecting output from every other encoder layer into the corresponding decoder layer. To match spatial dimensions, injected encoder features are combined with learnable masked tokens before channel-wise concatenation. The implementation details for the training and

Method	**FT**	**LP**	**KNN**
Contrastive representation learning			
MoCo-v3[45]	83.2	76.7	66.6
DINO[34]	83.3	**78.2**	**76.1**
Con MIM[273]	**83.7**	39.3	-
Generative representation learning			
Data2Vec[8]	84.2	68.0	33.2
I-JEPA[7]	-	**72.9**	-
CAE[42]	83.8	70.4	51.4
ADDP(VIT-L) [241]	**85.9**	23.8	-
Latent MIM[262]	83.0	72.0	50.1
MAE[98]	83.6	67.8	27.4
Ours ($\alpha_{\min}$ = 0.6)	82.9	**72.7**	**63.9**

Table 2.1: **Benchmark of representations on ImageNet-1K.** We evaluate learned features using standard evaluation protocols: linear probing (LP), fine-tuning (FT) and K-Nearest Neighbor (KNN). With only a simple architectural modification to MAE [98] and trained purely with pixel-wise reconstruction loss, we achieve 72.7% LP accuracy and 63.9% KNN accuracy, significantly narrowing down the gap between generative and contrastive representation learning frameworks.

evaluation are shown in Section A.1. He et al. [98] show the desired representations appear at the end of encoder; we therefore apply our decaying schema only to the encoder.

2.4.1 Representation Learning on ImageNet-1k

We follow the default hyperparameters from MAE [98] to pretrain ImageNet-1K train split [58] and use the standard protocol to evaluate the learned representation with end-to-end finetuning (FT), linear probing (LP) and K-Nearest Neighbour (KNN, K = 20), for image classification task. Please see the appendix for detailed experimental setups.

We report the results in Table 2.1. In the top half of the table, we present methods that employ a contrastive loss. Although these methods produce the best probing accuracies, their success depends on a carefully designed data augmentation process, which may need to be tuned for each different data distribution. In the bottom half, we show several methods based on generative architecture. Our method simply extends MAE by constructing an implicit feature bottleneck and shows significant improvements over the MAE baselines for both linear probing (72.7% *vs.* 67.3%) and

Feat. Dim.	Enc. Depth (L)	$\alpha_{\min}$					α_L^{eff}			
		0.6	0.7	0.8	0.9	1.0	(0, 1e-3)	[1e-3, 1e-2)	[1e-2, 1e-1)	[1e-1, 1]]
384	12	**78.5**	78.1	75.2	73.5	69.2	-	**78.5**	78.1	75.2
768	12	**83.6**	81.8	79.8	79.2	76.5	-	**83.6**	81.8	79.8
1024	12	**83.2**	82.5	82.1	79.3	78.0	-	**83.2**	82.5	82.1
768	18	83.5	**85.0**	84.4	81.8	79.2	78.5	**85.0**	84.4	81.8
1024	24	84.3	**86.0**	84.5	84.3	81.4	82.4	**86.0**	-	84.3

Table 2.2: **Linear Probing accuracy of MAE on ImageNet-100 varying $\alpha_{\min}$ and architecture size.** We show our method consistently improves the performance across all configurations. We notice the encoder depth rather than feature dimension influences the optimal $\alpha_{\min}$. We attribute this behavior to the scaling effect of the input data to encoder's final layer, quantified as $\alpha_L^{\text{eff}} = \prod_{l=0}^{L-1} \alpha_l$. Deeper models require larger $\alpha_{\min}$ to maintain a consistent cumulative decay effect and we find setting $\alpha_{\min}$ such that $\alpha_L^{\text{eff}} \in$ [1e-3, 1e-2) yield the best performance. With our strategy, a smaller model (768 feature dim + 12 layer) outperforms a bigger one (1024 feature dim + 24 layer) with standard residual connections.

$\alpha_{\min}$	α_l **scheduler**	**UNet**	**LP**
0.6	Linear	Yes	**83.6**
0.6	Linear	No	61.5
0.6	Cosine	Yes	82.8
0.7	Linear	Yes	81.8
0.7	Cosine	Yes	82.9
-	Learnable α_l	Yes	79.5

(a) **Effect of Skip Connections and α_l scheduler**. Skip connections are critical for performance. The choice of schedulers has less impact and learnable α_l is worse than pre-fixed α_l.

Configurations	**Decay Block**	$\alpha_{\min}$	**LP**
$\boldsymbol{x}_{l+1} = \boldsymbol{x}_l + f_{\theta_l}(\boldsymbol{x}_l)$	-	–	76.5
$\boldsymbol{x}_{l+1} = \boldsymbol{x}_l + \sqrt{0.5} f_{\theta_l}(\boldsymbol{x}_l)$	MLP & Atten.	–	76.9
$\boldsymbol{x}_{l+1} = \sqrt{0.5}\,(\boldsymbol{x}_l + f_{\theta_l}(\boldsymbol{x}_l))$	MLP & Atten.	–	82.6
$\boldsymbol{x}_{l+1} = \alpha_l \boldsymbol{x}_l + f_{\theta}(\boldsymbol{x}_l)$	Atten.	0.6	79.3
$\boldsymbol{x}_{l+1} = \alpha_l \boldsymbol{x}_l + f_{\theta}(\boldsymbol{x}_l)$	MLP	0.6	80.6
$\boldsymbol{x}_{l+1} = \alpha_l \boldsymbol{x}_l + f_{\theta}(\boldsymbol{x}_l)$	MLP & Atten.	0.6	**83.6**

(b) **Other Decay Schemas**. We conduct ablations using a variety of scalings of the residual connection and observe that our design produces the best results.

Table 2.3: **Ablation experiments of MAE using ViT-B/16 in ImageNet-100.** We report the results with linear probing (LP) accuracy.

K-Nearest Neighbour (63.9% *vs.* 27.4%), outperforming Data2Vec, Latent MIM and CAE and giving a probing accuracy competitive with I-JEPA, without needing to use explicit feature alignment.

End-to-end fine-tuning (FT), unlike linear probing which only trains a single linear layer, updates the entire network for image classification. Since the features can shift significantly from their pre-training state during end-to-end updating, we argue that this may not accurately reflect the quality of the learned representations. For example, DINO demonstrates superior performance in various downstream vision tasks compared to MAE, but its fine-tuning performance is worse than MAE. Similarly, ConMIM and ADDP exhibit poor linear probing performance, suggesting lower-quality representations, yet their fine-tuning performance surpasses that of contrastive learning methods. Nevertheless, we still provide the fine-tuning results for reference.

2.4.2 Ablation Studies on ImageNet-100

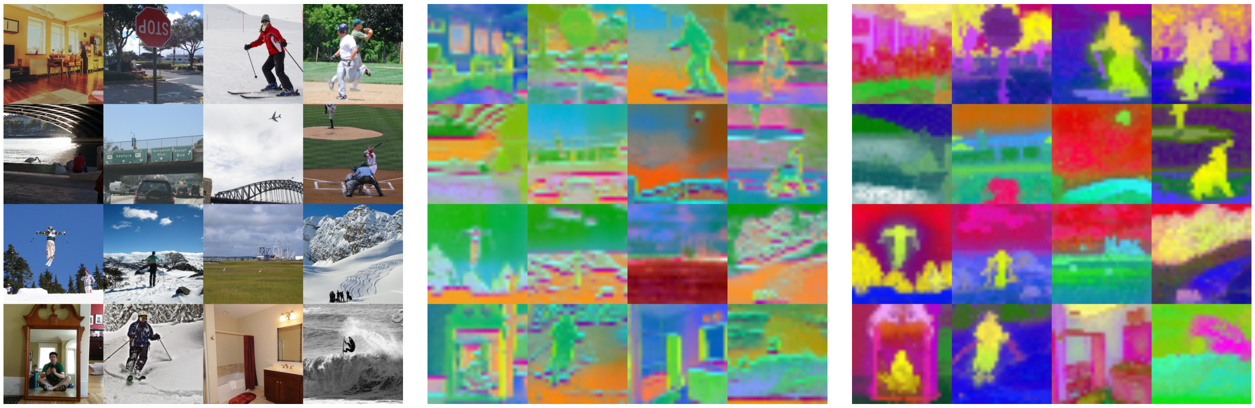
We conduct ablations on several properties of our framework on ImageNet-100. A summary of results can be found in Tables 2.2 and 2.3.

Decay rate $\alpha_{\min}$. The only parameter of our framework is $\alpha_{\min}$, the minimum scaling factor applied to the final layer. In Table 2.2,we show linear probing scores for varying values of $\alpha_{\min}$. We observe that $\alpha_{\min} \in [0.6, 0.7]$ works well for most cases. If $\alpha_{\min}$ is too small, for example, $\alpha_{\min} \leq 0.4$, we observe that the training becomes unstable.

Architecture size. In Table 2.2, we also run experiments over multiple architecture choices. We notice encoder depth L rather than feature dimension that influences the optimal choices of $\alpha_{\min}$ and deeper models requires larger $\alpha_{\min}$. We attribute this phenomenon to the cumulative decaying effect of the final layer, quantified as $\alpha_L^{\text{eff}} = \prod_{l=1}^{L} \alpha_l$. Heavy decaying would harm the optimization and selecting $\alpha_{\min}$ such that $\alpha_L^{\text{eff}} \in$ [1e-3, 1e-2) yields the best performance.

Skip connections. Another critical design choice in our network is to include skip connections that are not in the original MAE. As discussed in Section 2.3.2, if the MAE does not use skip connections, the bottleneck layer must preserve all information to reconstruct the input image accurately. This is opposed to learning abstract representations at bottleneck. These contrary effects significantly degrade the representation learned by the model, leading to a 22.1$\%$ drop in the linear probing score, as we report in Table 2.3a.

α_l **Scheduler.** We use linear scheduler $\alpha_l = 1 - \frac{(1-\alpha_{\min})}{L} l$ as a default choice. In table 2.3a, we also experiment with a cosine scheduler but find it leads to worse performance for $\alpha_{\min} = 0.6, 0.7$. Besides using a prefixed α_l scheduler, we also experiment with a learnable α_l, which resembles the setup of highway network [231]. We show the learnable α_l at each layer in Table A.1, appendix.



(a) Input Image (b) MAE (4.1 mIoU) (c) Ours (10.4 mIoU)

Figure 2.3: **Visualize learned representations using Zhang et al. [291] without cherry-picking.** We project the learned representations onto a 3-channel feature map, visualized as RGB images. Our method learns more abstract and semantically consistent representations compared to the baseline MAE. This visual comparison is further supported by benchmarking on unsupervised semantic segmentation tasks, where our approach achieves better results (10.4 mIoU) compared to the baseline MAE (4.1 mIoU).

From the table, we don't find consistent patterns over network depth and the performance is worse than our predefined α_l scheduler.

Different decay schema. We also explore decay schema, with results summarized in Table 2.3b: (1) Scaling both branches of the residual blocks simultaneously by applying a constant factor, $\alpha = \sqrt{0.5}$, to both $\boldsymbol{x}$ and $f_{\theta_l}(\boldsymbol{x})$. (2) Scaling only f_{θ_l} using the same constant factor, $\alpha = \sqrt{0.5}$. (3) Applying our proposed schema exclusively to either the attention or MLP branch.

Among these, (2) shows no significant improvement over the baseline, while (1) yields some improvement but still underperforms compared to our approach. By analyzing (1) and (2), we demonstrate that the representation gains are due to down-weighting the skip connection branch. Notably, recent diffusion models [125, 228] have employed (1) in their smaller convolutional neural network but don't provide systemic analysis. However, applying decay only to the MLP or attention branch reduces the overall decaying effect across the network, resulting in lower performance compared to our schema, which achieves the best performance among the tested designs.

2.4.3 Embdding analysis

We qualitatively evaluate the feature learning in Figure 2.3 and we adopt the pixel-wise embedding approaches proposed by Zhang et al. [291] and also described in Chapter 4 to group the representations from the last layer of the encoder into a lower dimensional space. We use their default hyperparameters to cluster representations across COCO validation set and render the top 3 eigenvectors as RGB channels of images. From the visualization, ours learns abstract representation and the object from the same categories have similar color, indicating a global consistent semantic grouping. The baseline MAE, on the other hand, doesn't show clearly global semantic patterns.

We also benchmark the clustering quantitatively, following the postprocessing protocol [291] to produced unsupervised semantic segmentation and report the results as the mean intersection of union (mIoU). Ours (10.41 mIoU) achieves 6.31 mIoU improvement over the baseline (4.10 mIoU), which supports the visual comparison.

2.5 Experiments on Diffusion Models

Diffusion models. We use U-ViT [9], a ViT-based diffusion model with skip connections between the encoder and decoder, as the baseline for our diffusion model experiments. Recent studies [272, 11] suggest that diffusion models learn the best semantic representations near the decoder's latter stages. Therefore, we apply our proposed decay mechanism up to the end of the decoder. While this design might be suboptimal, as the smallest decay factor may not align with the layers holding the best semantic representations, we demonstrate in practice that this simple approach effectively enhances both the learned representations and the quality of generated outputs.

Experimental details. We utilize the default scheduler and sampler from U-ViT [9], replacing only the residual connections with our proposed decayed shortcut connections. We train unconditional diffusion models on CIFAR-100 and ImageNet-100 without using image class labels. Additionally, we train a class-conditional diffusion model on ImageNet-100 to validate our design

	Linear Probing (Acc.)↑				**Generation quality (FID)**↓			
α_{min} Dataset	1.0	0.8	0.7	0.6	1.0	0.8	0.7	0.6
CIFAR-100 (Uncon.)	62.47	63.58	**66.86**	64.63	14.34	11.65	**8.99**	11.71
ImageNet-100 (Uncond.)	72.8	74.5	**76.1**	75.8	44.40	**40.96**	41.17	43.51
ImageNet-100 (Class Cond.)	-	-	-	-	6.93	5.75	5.11	**4.98**

Table 2.4: **Performance using diffusion models.** In diffusion models, we demonstrate that our proposed decayed identity shortcut (with $\alpha_{\text{min}} < 1.0$) enhances probing accuracy and improves generation quality across various datasets and configurations in both classes conditional and unconditional setups.

across different tasks. For ImageNet-100, instead of training directly on pixels, we adopt a latent diffusion [204] approach by running the model in the latent space of a pretrained VAE, which reduces input resolutions from 256x256x3 to 32x32x4. We use U-ViT-Mid for ImageNet-100 and U-ViT-small for CIFAR-100. For model and training details, please refer to Bao et al. [9].

We evaluate the learned representations with linear probing and we train a linear classifier over the frozen representations. We report the results as the best configurations, including the choices of layer index and noise level, that yields the best performance

Results. Our results are presented in Table 2.4, where we demonstrate that replacing residual connections with our proposed decayed identity shortcuts consistently enhances representation quality and image generation across both datasets and tasks (conditional and unconditional generation). Notably, this improvement is achieved without introducing any additional learnable parameters. We provide the visualization of generated images in the appendix for qualitative comparisons (Figure A.4).

2.6 Discussion on feature rank

In this section, we try to answer a key question: How and why do residual connections impact the abstraction level of the deeper layers in a neural network? We delve deeper into how our design reinforces the low-rank bias of neural networks and try to connect our method to ideas in existing works [113]. To this end, we visualize the training dynamics of our method and analysis the feature rank of our approach to provide a holistic analysis.

Low-rank simplicity bias. Huh et al. [113] investigate the low-rank simplicity bias in deeper feed-forward neural networks, which drives neural networks to find low-rank solutions. At the same time, they make an empirical observation that deeper residual networks do not show a similar rank contracting behavior.

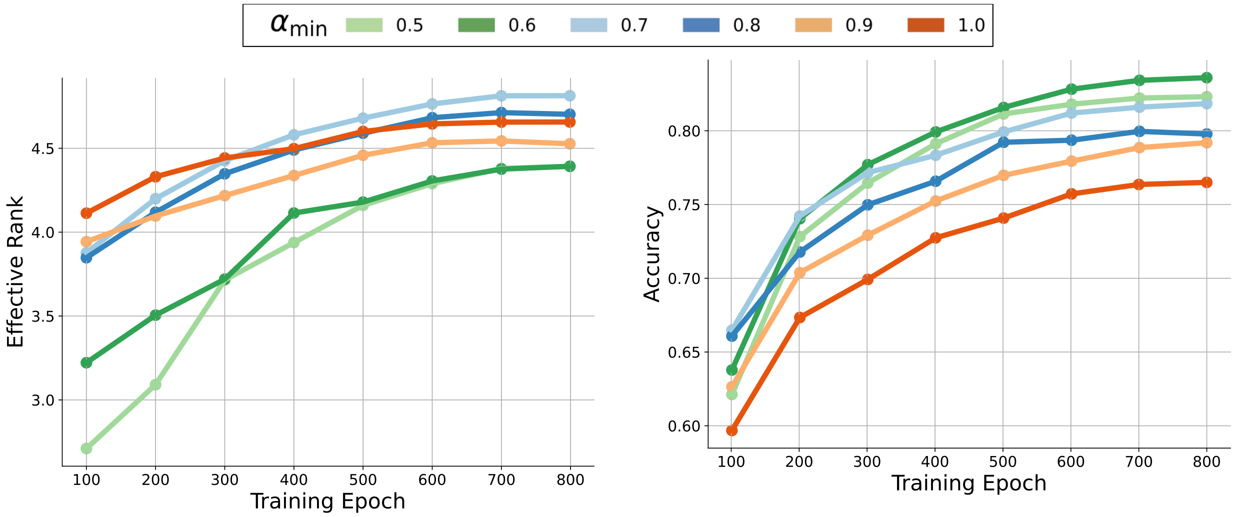
Effective rank. For analysis purpose, Huh et al. [113] quantify the rank of the learned representation using the *effective rank*, which is a continuous measure. For a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, the effective rank $\rho(\boldsymbol{A})$ is defined as the Shannon entropy of the normalized singular values [207]:

$$\rho(\boldsymbol{A}) = -\sum_{i}^{\min(n,m)} \bar{\sigma}_i \log \bar{\sigma}_i, \tag{2.6}$$

where $\bar{\sigma}_i = \sigma_i / \sum_j \sigma_j$ denotes the i^{th} normalized singular value. Intuitively, $\rho(\boldsymbol{A})$ is small when a few singular values dominate and large when singular values are evenly spread, hence giving a good continuous approximation for matrix rank. In the following subsections, we use the singular values from the covariance matrix $\boldsymbol{A}_\theta$ of the last-layer features to compute $\rho(\boldsymbol{A})$, where $\boldsymbol{A}_\theta(i, j)$ denotes the covariance of the learned class tokens for the i^{th} and j^{th} samples.

Inspired by Huh et al. [113], *we conjecture that the improvement to feature learning capability of our method can mainly be attributed to the decayed identity shortcuts promoting low-rank features at the bottleneck.* In Figures 2.4a and 2.4b, we measure the training dynamics of the models (feature dimension 768 and encoder depth 12) in terms of accuracy and the effective rank, for different values of α_{min}.

During the early epochs, models with lower α_{min} tend to exhibit both lower effective rank and higher probing accuracy, supporting our hypothesis. As training progresses, the correlations between α_{min} and effective rank become less precise. We suspect this is due to the model's effort to compensate for large decay factor. Despite this, we can still conclude that lower $\alpha_{\text{min}} \in [0.5 - 0.6]$ results in lower feature rank and better probing accuracy compared to higher $\alpha_{\text{min}} \in [0.7 - 1.0]$.



(a) Effective Rank of MAE over training epochs.

(b) Linear probing accuracy of MAE over training epochs.

Figure 2.4: For MAE pretrained on ImageNet-100, we present visualizations of (a) the training dynamics of the effective rank for different values of α_{min}, (b) the linear probing accuracy for various α_{min}, demonstrating that a lower effective feature rank is associated with better performance.

Compatibility with contrastive learning frameworks. Despite substantial improvements from applying our decayed identity shortcuts in generative models, we note that our approach does not easily extend to contrastive learning frameworks, where the low rank inductive bias conflicts with the training objectives, e.g., MoCov3 [45] include an universal repulsive term in the denominator to increase the feature rank. Rankme [79] confirms this by showing conservative learning models with higher feature ranks yield better results.

2.7 Conclusion

Huh et al. [113] raise a key insight in their work – that how a neural network is parameterized matters for fitting the data – and investigate the inductive low-rank bias of stacking more linear layers in a network. In this work, we observe that the ubiquitous residual network [95] may not be the ideal network parametrization for representation learning and propose a modification of the shortcut path in residual blocks that significantly improves unsupervised representation learning. We explore the connection between our reparameterization of the residual connection and the effective rank of the learned features, finding a correlation between good representations and low-rank representations.

Our work calls into question a fundamental design choice of neural networks that has been used in many modern architectures. By rethinking this choice, the door is open for further reparametrizations and improvements to unsupervised representation learning. The results we show provide a prompt for more extensive investigations into the connection between low effective rank and high-quality abstract representations, as well as the exploration of underlying theoretical mechanisms for this relationship.

CHAPTER 3
STRUCTURAL ADVERSARIAL OBJECTIVES FOR SELF-SUPERVISED REPRESENTATION LEARNING

Within the framework of generative adversarial networks (GANs), we propose objectives that task the discriminator for self-supervised representation learning via additional structural modeling responsibilities. In combination with an efficient smoothness regularizer imposed on the network, these objectives guide the discriminator to learn to extract informative representations, while maintaining a generator capable of sampling from the domain. Specifically, our objectives encourage the discriminator to structure features at two levels of granularity: aligning distribution characteristics, such as mean and variance, at coarse scales, and grouping features into local clusters at finer scales. Operating as a feature learner within the GAN framework frees our self-supervised system from the reliance on hand-crafted data augmentation schemes that are prevalent across contrastive representation learning methods. Across CIFAR-10/100 and an ImageNet subset, experiments demonstrate that equipping GANs with our self-supervised objectives suffices to produce discriminators which, evaluated in terms of representation learning, compete with networks trained by contrastive learning approaches.

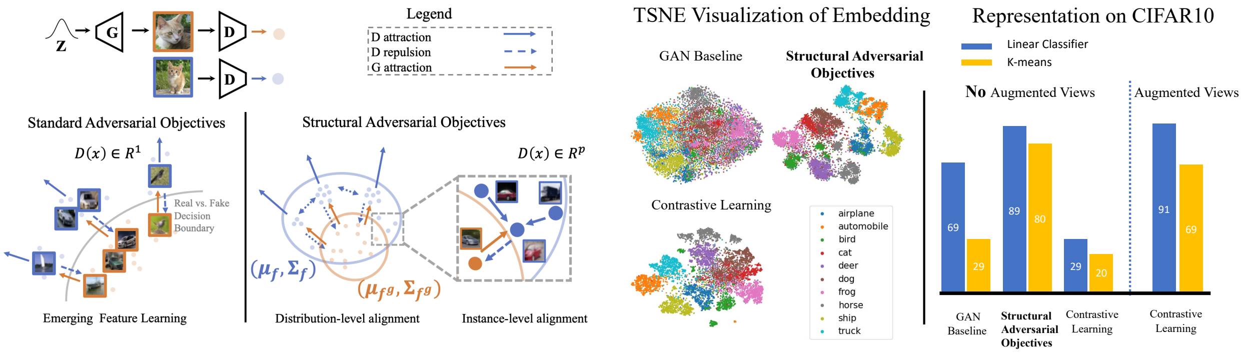
3.1 Introduction

Unsupervised feature learning algorithms aim to directly learn representations from data without reliance on annotations, and have become crucial to efforts to scale vision and language models to handle real-world complexity. Many state-of-the-art approaches adopt a contrastive self-supervised framework, wherein a deep neural network is tasked with mapping augmented views of a single example to nearby positions in a high-dimension embedding space, while separating embeddings of different examples [266, 97, 40, 43, 87, 282]. Though requiring no annotation, and hence unaffected by assumptions baked into any labeling procedure, the invariances learned by these models are still influenced by human-designed heuristic procedures for creating augmented views.

The recent prominence of contrastive approaches was both preceded by and continues alongside a focus on engineering domain-relevant proxy tasks for self-supervised learning. For computer vision, examples include learning geometric layout [60], colorization [286, 149], and inpainting [189, 98]. Basing task design on domain knowledge may prove effective in increasing learning efficiency, but strays further from an alternative goal of developing truly general and widely applicable unsupervised learning techniques.

Another family of approaches, coupling data generation with representation learning, may provide a path toward such generality while also escaping dependence upon the hand-crafted elements guiding data augmentation or proxy task design. Generative adversarial networks (GANs) [85] and variational autoencoders (VAEs) [136] are prime examples within this family. Considering GANs, one might expect the discriminator to act as an unsupervised representation learner, driven by the need to model the real data distribution in order to score the generator's output. Indeed, prior work finds that some degree of representation learning occurs within discriminators in a standard GAN framework [194]. Yet, to improve generator output quality, limiting the capacity of the discriminator appears advantageous [5] – a choice potentially in conflict with representation learning. Augmenting the standard GAN framework to separate encoding and discrimination responsibility into different components [64, 69], along with scaling to larger models [63], are promising paths forward.

However, it has been unclear whether the struggle to utilize vanilla GANs as effective representation learners stems from inherent limitations of the framework. We provide evidence to the contrary, through an approach that significantly improves representations learned by the discriminator, while maintaining generation quality and operating with a standard pairing of generator and discriminator components. To enhance GANs into effective representation learners, our approach need only modify the training objectives within the GAN framework. Our contributions are as follows:



(a) Standard *vs.* proposed structural adversarial objectives for feature learning

(b) Visualizing and quantitatively evaluating discriminator features on CIFAR-10

Figure 3.1: *(a) Structural GAN Objectives:* In a standard GAN, the discriminator produces a scalar score to discern real and fake samples. As the generator improves, representations produced by the discriminator will update structurally similar data in a similar direction, displayed as solid blue arrows. Our structural adversarial objectives enhance such learning capability by optimizing the feature vectors produced by the discriminator. We achieve this by manipulating mean and variance at a coarser scale and implementing instance-level grouping at a finer scale, allowing the discriminator to explicitly learn semantic representations, in addition to distinguishing between real and fake. *(b) Discriminator as Semantic Representation Learner:* Trained with our new objectives, the discriminator's learned feature embedding reveals category semantics and achieves performance competitive with contrastive learning methods. Unlike self-supervised contrastive methods, our approach *does not* depend upon learning from different views obtained via a data augmentation scheme.

- We propose adversarial objectives resembling a contrastive clustering target (Figure 3.1). These self-supervised objectives prompt the discriminator to learn semantic representations, without depending on data augmentation to fuel the learning process.

- We introduce an effective regularization approach that utilizes the approximation of the spectral norm of the Jacobian to regulate the smoothness of the discriminator. This methodology enables the discriminator to strike a balance between its capacity to learn features and its ability to properly guide the generator.

- On representation learning benchmarks, our method achieves competitive performance with recent state-of-the-art contrastive self-supervised learning approaches, even though we do not leverage information from (or even have a concept of) an augmented view. We demonstrate that supplementing a GAN with our proposed objectives not only enhances the discriminator as a representation learner, but also improves the quality of samples produced by the generator.

3.2 Related Work

3.2.1 Generative Feature Learning

GANs [85] include two learnable modules: a generator $\boldsymbol{G}$, which produces synthetic data given a sample $\boldsymbol{v}$ from a prior, and a discriminator $\boldsymbol{D}$, which learns to differentiate between the true data $\boldsymbol{x}$ and generated samples $\boldsymbol{G}(\boldsymbol{v})$. Here, θ, ϕ denote the trainable parameters. During training, $\boldsymbol{G}$ and $\boldsymbol{D}$ are alternatively updated in an adversarial fashion, which can be formulated as a minimax problem:

$$\min_{\boldsymbol{G}} \max_{\boldsymbol{D}} \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[\log \boldsymbol{D}(\boldsymbol{x})] - \mathbb{E}_{\hat{\boldsymbol{x}}\sim \boldsymbol{G}(\boldsymbol{v})}[1 - \log \boldsymbol{G}(\hat{\boldsymbol{x}})]. \tag{3.1}$$

Much research on GANs has focused on improving the quality of generated data, yielding significant advances [125, 126, 128, 129, 213, 56]. Other efforts have focused on evolving capabilities, including conditional and controllable generation, *e.g.,* text-guided [283, 103] or segmentation-guided [297, 39] generation. In comparison, adopting GANs for unsupervised feature learning has been more scarcely explored. In this area, an adversarial approach dependent upon an additional encoder component [64, 69, 63, 115] appears most successful to date. Here, the encoder is tasked to invert the generator with a discriminator acting on (data, latent) pairs and representation learning is the responsibility of the encoder, rather than the discriminator.

Besides GANs, other generative models also demonstrate feature learning capability. Recent efforts [285, 171] discard the low-level structures in VAE and Flow models to improve learned representations. Du et al. [68] shows that an unsupervised energy model can learn semantic structures, *e.g.,* segmentation and viewpoint, from images. Preechakul et al. [193] attach an encoder to a diffusion model and show that it learns high-level feature representations. We adopt an orthogonal approach that, by imposing structural adversarial objectives in GAN training, tasks the discriminator to learn richer data representations.

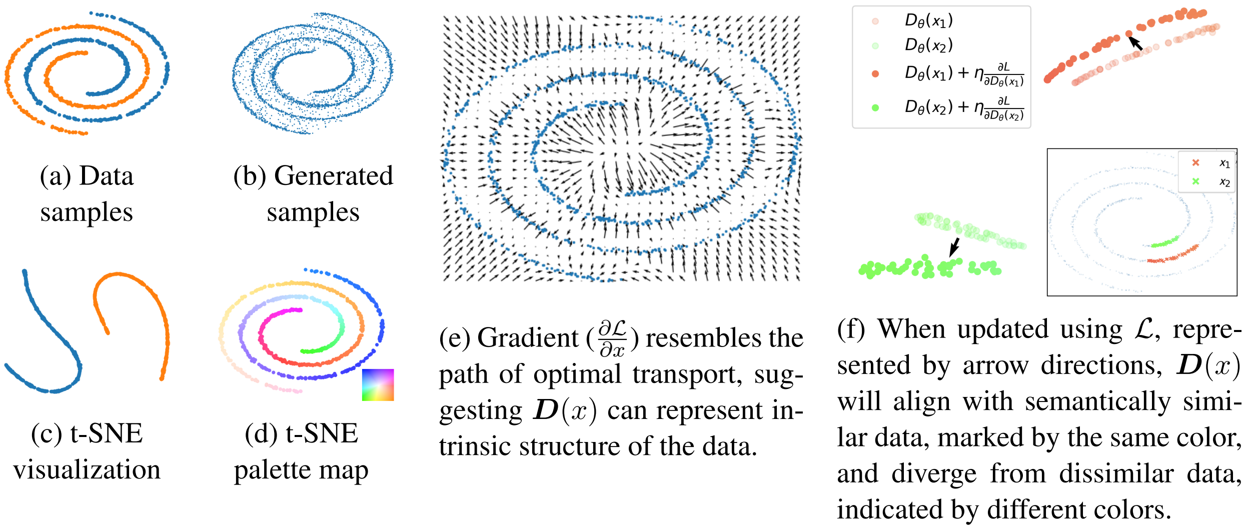
3.2.2 Contrastive Self-Supervised Learning

Self-supervised learning with a contrastive approach has shown enhanced feature learning capability and has evolved to nearly match the performance of its supervised counterparts. From initial impactful results in vision and language [266, 97, 40, 195], this technique has recently been employed across a variety of domains [118, 144, 90]. A popular strategy involves using a Siamese architecture to optimize the InfoNCE objective, which aims to maximize the feature similarity across augmented views, while repulsing from all other instances to maintain feature uniformity [266, 97, 40, 186]. Another strategy simplifies this pipeline by dropping the negative terms and leveraging specific architectural designs to prevent collapsed solutions [43, 87]. As an alternative to operating on an l_2 normalized embedding, other approaches [33, 34, 252] enforce clustering consistency across views. Inspired by masked language modeling, He et al. [98] and Bao et al. [10] propose variants in the image domain by tasking an autoencoder to predict masked pixels.

Though contrastive approaches yield strong benchmark results, Tian et al. [243] showcase the limitations of view-invariant assumptions and demonstrate their sensitivity to the parameters of augmentation schemes. Zhang and Maire [287] raise a concern with applying these methods to broader unconstrained datasets, where multiple object instances within the same image should not have mutually invariant representations.

3.2.3 Stabilizing GAN Training

Despite the ability to generate high-quality samples, successfully training GANs remains challenging due to the adversarial optimization. Several approaches have been proposed to stabilize training and enable scaling to larger models. Heusel et al. [102] suggest maintaining separate learning rates for the generator and discriminator, in order to maintain local Nash equilibrium. Arjovsky et al. [5]



(a) Data samples

(b) Generated samples

(c) t-SNE visualization

(d) t-SNE palette map

(e) Gradient ($\frac{\partial \mathcal{L}}{\partial x}$) resembles the path of optimal transport, suggesting $\boldsymbol{D}(x)$ can represent intrinsic structure of the data.

(f) When updated using $\mathcal{L}$, represented by arrow directions, $\boldsymbol{D}(x)$ will align with semantically similar data, marked by the same color, and diverge from dissimilar data, indicated by different colors.

Figure 3.2: We train a GAN with our structural objectives on a synthetic *double spiral* dataset. We show: *(a) training data* color-coded based on ground truth assignments; *(b) generated samples*; *(c,d) learned representations* visualized by t-SNE [247], and colored according to ground truth categories *(c)* as well as a 2D palette map *(d)*. Additionally, we highlight *(e) structural correspondence* of $\boldsymbol{D}(x)$ via $\frac{\partial \mathcal{L}}{\partial x}$, and in *(f)*, we visualize $\boldsymbol{D}(\boldsymbol{x})$ using t-SNE, showcasing the emerging capability for learning semantic features induced by our loss $\mathcal{L}$ (Eqn. 3.8).

and Gulrajani et al. [91] consider constraining the discriminator's Lipschitz constant with gradient clipping and gradient norm penalization. In contrast to regularizing model-wise functionality, Miyato et al. [179] implement layer-wise spectral normalization schemes by dividing parameters with their leading singular value, which is widely adopted in recent state-of-the-art models. Wu et al. [264] and Bhaskara et al. [20] instead propose to build a Lipschitz-constrained function by dividing the output with the gradient norm, and show it can preserve model capacity. However, none of these methods suit our case, since spectral normalization [179] harms model capacity, and gradient-based regularization only works for scalar output, limiting the use of structural objectives.

3.3 Method: Feature Learning with the Discriminator

Our goal is to task $\boldsymbol{D}$ as both a discriminator and a feature extractor that learns semantic representations of real data. We motivate this design from empirical observations of GAN discriminator behavior. Figure 3.1a conveys some intuition behind our design, while Figure 3.2 illustrates results, as well as discriminator learning dynamics when applying our method to a synthetic dataset.

As Figure 3.2f shows, the updating direction induced by our loss enables $\boldsymbol{D}(\boldsymbol{x})$ to position example $\boldsymbol{x}$ close to similarly structured examples while diverging away from dissimilar ones. Such behavior is not necessarily limited to our system; we hypothesize that it arises in broader contexts due to a Lipschitz-regularized discriminator producing gradients that rearrange the embedding along an optimal transport path, as shown in Figure 3.2e. As a consequence, structurally similar samples will be updated in a similar direction. Tanaka [237] establishes this idea in the context of Wasserstein GANs [5].

This conjecture suggests that, in a standard GAN, the discriminator implicitly learns some, but perhaps not all, aspects of a semantic representation. We are therefore motivated to propose explicit objectives for the discriminator that are both compatible with its original purpose (providing informative gradients to the generator) and that require it to produce an embedding that captures additional semantic structure of the data distribution.

3.3.1 Structural Adversarial Objectives

Instead of producing a scalar output, we architect $\boldsymbol{D}$ to learn the mapping from the data space to the feature space, $\boldsymbol{D} : \mathcal{X} \rightarrow \mathcal{Z}$. We denote the output from $\boldsymbol{D}$ on real data and fake (generated) samples as $\boldsymbol{z}$ and $\boldsymbol{z}^g$, respectively. Here $\boldsymbol{z}, \boldsymbol{z}^g \in \mathrm{S}^{p-1}$ are normalized and live in a unit hypersphere. We also maintain unnormalized counterparts $\tilde{\boldsymbol{z}}$ and $\tilde{\boldsymbol{z}}^g$ of $\boldsymbol{z}$ and $\boldsymbol{z}^g$; Section 3.3.2 explores their utility.

Driving the formulation of our proposed objectives is the idea to require $\boldsymbol{D}$ to model the real and fake distributions (without collapse), while $\boldsymbol{G}$ adversarially attempts to align these distributions. As related prior work, OT-GAN [211] proposes explicit optimal-transport adversarial objectives for this purpose, but requires a large batch size (8K) to stabilize. Instead, our objectives operate hierarchically and regularize the learned embeddings at two levels of granularity:

(1) At a coarse level, we align the distribution statistics of the discriminator, focusing on its mean

and covariance: $\boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\mu}_{\boldsymbol{z}^g} \in \mathbb{R}^p$ and $\boldsymbol{\Sigma}_{\boldsymbol{z}}, \boldsymbol{\Sigma}_{\boldsymbol{z}^g} \in \mathbb{R}^{p \times p}$. Here, we simplify the optimization by assuming a diagonal structure of the covariance matrix. This enables efficient alignment of the two distributions with tolerance to finer-grained differences.

(2) At a finer level, we focus on reorganizing embeddings by constructing clusters using local affinity. The corresponding objective tasks $\boldsymbol{D}$ with learning local geometry, further focusing the GAN on feature alignment between real and fake distributions.

Coarse-scale optimization by aligning distributions. To align the distributions in terms of mean and covariance, we can employ a distance function $d(\cdot)$ and optimize the minimax objective:

$$\mathcal{L}_{\text{Gaussian}} := \min_{\boldsymbol{G}} \max_{\boldsymbol{D}} d(\boldsymbol{z}, \boldsymbol{z}^g). \tag{3.2}$$

One widely adopted candidate for $d(\cdot)$ is Jensen-Shannon divergence (JSD) due to its symmetry and stability. For two arbitrary probability distributions P, Q, JSD admits the following form:

$$\text{JSD}(P||Q) = \ \tfrac{1}{2}(D_{\text{KL}}(P||\tfrac{P+Q}{2}) + D_{\text{KL}}(Q||\tfrac{P+Q}{2})) = \tfrac{1}{2}(H(\tfrac{P+Q}{2}) - \tfrac{1}{2}\left(H(P) + H(Q)\right)) \tag{3.3}$$

where D_{KL}, H denote Kullback-Leibler divergence and entropy, respectively. We can compute entropy for Q, P using closed-form expressions. However, entropy for $(P + Q)/2$ is difficult to compute exactly and generally requires Monte Carlo simulation, an infeasible computational approach in high dimensional space. To tackle this problem, we follow Hershey and Olsen [101] to approximate $\frac{P+Q}{2}$ by a single Gaussian and estimate sample mean and covariance by joint samples of P and Q, which yields an upper bound of $H(\frac{P+Q}{2})$; the bound is tight when $P = Q$. Putting these together, we obtain our distance function for the coarser scale objective[1]:

$$\text{JSD}(\boldsymbol{z}, \boldsymbol{z}^g) \approx \log \frac{\det \boldsymbol{\Sigma}_{\boldsymbol{z}+\boldsymbol{z}^g}}{\sqrt{det \boldsymbol{\Sigma}_{\boldsymbol{z}} \det \boldsymbol{\Sigma}_{\boldsymbol{z}^g}}}. \tag{3.4}$$

[1] Note that though Eqn. 3.4 and MCR in Dai et al. [56] are constructed similarly, the latter is interpreted from a coding rate reduction perspective.

Another well-established metric between two Gaussian distributions is Bhattacharyya distance D_B:

$$D_B(\boldsymbol{z}, \boldsymbol{z}^g) := \frac{1}{8}(\boldsymbol{\mu}_{\boldsymbol{z}} - \boldsymbol{\mu}_{\boldsymbol{z}^g})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_{\boldsymbol{z}} - \boldsymbol{\mu}_{\boldsymbol{z}^g}) + \frac{1}{2}\log\frac{\det\boldsymbol{\Sigma}}{\sqrt{\det\boldsymbol{\Sigma}_{\boldsymbol{z}}\det\boldsymbol{\Sigma}_{\boldsymbol{z}^g}}}, \tag{3.5}$$

where $\boldsymbol{\Sigma} = \frac{\boldsymbol{\Sigma}_{\boldsymbol{z}}+\boldsymbol{\Sigma}_{\boldsymbol{z}^g}}{2}$. Though having different geometric interpretations, it is notable that D_B and JSD have similar format and, when maximizing $d(\boldsymbol{z}, \boldsymbol{z}^g)$ for $\boldsymbol{z}$, both aim to uniformly repulse z to prevent producing collapsed representations. In experiments, we observe that these two distances yield similar performance and we use JSD as our default choice for $d(\cdot)$ since it has a slightly faster convergence rate and yields better quality for generated images.

Fine-grained optimization via clustering. We perform mean-shift clustering on $\boldsymbol{z}$ by grouping nearby samples. We simplify the clustering process by equally averaging each neighbor sample, rather than using feature similarity to reweight their contribution. To improve nearest neighbor search stability, we maintain a rolling updated memory bank $\boldsymbol{z}^m$ that stores the embedding of all real images as a query pool and use the backbone representation $\boldsymbol{z}^b$, rather than $\boldsymbol{z}$, as the key to computing feature similarity. Denoting $\{\boldsymbol{z}_{i,j}\}_{j=1}^k$ and $\{\boldsymbol{z}_{i,j}^g\}_{j=1}^K$ as the returned K nearest neighbors of real images embedding for $\boldsymbol{z}_i$ and $\boldsymbol{z}_i^g$ respectively, our clustering objective is:

$$\mathcal{L}_{\text{cluster}} := \max_{\boldsymbol{D}} \frac{1}{NK}\sum_{i=1}^{N}\sum_{j=1}^{K} \boldsymbol{z}_{i,j}^\top \boldsymbol{z}_i + \min_{\boldsymbol{D}}\max_{\boldsymbol{G}} \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} {\boldsymbol{z}_{i,j}^g}^\top \boldsymbol{z}_i^g. \tag{3.6}$$

IC-GAN [35] implements a similar instance-wise objective. However, they use frozen embeddings from an off-the-shelf model rather than jointly learn an embedding, and their motivation is to improve image generation quality rather than learn semantic features — entirely different from our aim.

3.3.2 Smoothness Regularization

Besides reformulating adversarial targets for representation learning, we address another common issue in GAN training: balancing the discriminator's capacity and the smoothness constraint. Recent studies demonstrate that regularizing $\boldsymbol{D}$'s smoothness, or its Lipschitz constant, is critical for scaling GANs to large network architectures. Consider a continuous function $F : \mathbb{R}^m \rightarrow \mathbb{R}^p$. We can bound its Lipschitz constant by the spectral norm of Jacobian $\boldsymbol{J}_F$:

$$\|\boldsymbol{J}_F(\boldsymbol{x})\|_2 \leq \mathrm{Lip},$$

where $\|\cdot\|_2$ denotes the matrix spectral norm. However, computing the full Jacobian matrix is highly inefficient in standard backpropagation process since each backpropagation call can only compute a single row of the Jacobian matrix, which is impracticable as we usually need large embedding dimension p.

Algorithm 1: Approximating $\|\boldsymbol{J}_F(\boldsymbol{x})\|_2$ with power iterations

```
Input: Function $F : \mathbb{R}^m \rightarrow \mathbb{R}^p$; Stop gradient operator $sg(\cdot)$; Power iteration
  steps S; Batch size $b$; Input data $\boldsymbol{x} \in \mathbb{R}^{b \times m}$;
Init random vector $\boldsymbol{u} \sim \mathcal{N}(0,1) \in \mathbb{R}^{b \times p}$
for iter $= 1 \ldots S$ do
    $\boldsymbol{v} = \boldsymbol{u}\boldsymbol{J}_F(\boldsymbol{x}) / \|\boldsymbol{u}\boldsymbol{J}_F(\boldsymbol{x})\|_2$    //VJP
    $\boldsymbol{u} = \boldsymbol{J}_F(\boldsymbol{x})\boldsymbol{v} / \|\boldsymbol{J}_F(\boldsymbol{x})\boldsymbol{v}\|_2$    //JVP
end
Return: $\|\boldsymbol{J}_F(\boldsymbol{x})\|_2 \approx sg(\boldsymbol{u})\boldsymbol{J}_F(\boldsymbol{x})sg(\boldsymbol{v})$
```

Therefore, we propose to efficiently approximate $\|\cdot\|_2$ using power-iterations. Leveraging the fact that power-iteration is a matrix-free method, we do not need to explicitly compute the Jacobian matrix. Instead, we only need to access the matrix by evaluating the matrix-vector product, which can be efficiently computed by batch-wise VJP and JVP (Jacobian-Vector-Product) subroutine. Algorithm 1 presents the details, where only $(2S + 1)$ backpropagation calls are required to approximate $\|\boldsymbol{J_D}(\boldsymbol{x})\|_2$. In experiments, we find that $S = 1$ suffices for a ResNet-18 model.

We observe that maintaining $\|\tilde{\boldsymbol{z}}\|$ at a regular level benefits training stability. To this end, we use hinge loss to regularize the embedding norm and empirically observe it performs better than removing the hinge. Therefore, our smoothness regularization is:

$$\min_{\boldsymbol{D}} \mathbb{E}_{\boldsymbol{x}} \|\boldsymbol{J_D}(\boldsymbol{x}) - \mathrm{Lip}\|_2 + \lambda_h \mathbb{E}_{\tilde{\boldsymbol{z}}} \| \max(\|\tilde{\boldsymbol{z}}\| - 1, 0)\|_2 \tag{3.7}$$

where λ_h denotes the ratio for hinge regularization, and Lip denotes the Lipschitz target of $\boldsymbol{D}$, which is set to 1 by default. Unlike a layer-wise normalization scheme, *e.g.,* Spectral Norm [179], where demanding local regularization hurts the model's capacity, our proposed regularization scheme allows the network to simultaneously fit multiple objectives, *i.e.,* representation learning and smoothness regularization. The model does not have to sacrifice capacity for smoothness. Another benefit of our method is that our proposed term can work with the normalization layer. Spectral Norm cannot, because of the data-dependent scaling term in its normalization layer.

Overall training objective. We define our final objective as:

$$\mathcal{L} := \mathcal{L}_{\text{Gaussian}} + \lambda_c \mathcal{L}_{\text{cluster}} + \lambda_s \mathcal{L}_{\text{reg}}, \tag{3.8}$$

where λ_c, λ_s control the relative loss weights.

3.4 Experimental Settings

We train our model for 1000 epochs on CIFAR-10/100 and 500 epochs on ImageNet-10. We use the AdamW optimizer [169] with a constant learning rate of 2e-4 for both generator and discriminator. We additionally add 0.1 weight decay to the discriminator. We use batch size 500 on CIFAR-10/100 and 320 on ImageNet-10. We run a small-scale parameter tuning experiment for hyperparameters and find that setting $\lambda_h = 4, \lambda_c = 3, \lambda_s = 5$ yields the best result. For simplicity, we run a single discriminator update before optimizing the generator, *i.e.,* $n_{dis} = 1$.

As a widely adopted GAN training trick, we maintain a momentum-updated discriminator and generator for evaluation purposes and find they produce stable data representations and better quality images. We also try producing $\boldsymbol{z}^b$ from momentum models for nearest neighbor searching, which slightly improves performance in all benchmarks. We set the memory bank size $|\boldsymbol{z}^m| = 10240$, which is smaller than all datasets, preventing the model from accidentally picking features from augmented versions of the input image. Appendix B.1 provides more model configuration details.

3.5 Results And Discussion

3.5.1 Synthetic Data

For illustrative purposes, we first train a GAN using our structural objectives on the synthetic double spirals dataset [158]. Here, we implement discriminator and generator as multi-layer perceptrons and keep all other configuration, *e.g.,* normalization layers, activation functions, objectives, and learning rate, consistent with our settings for experiments on real images.

Figure 3.2a demonstrates that the generated samples capture all data modes, with few outlier samples between spirals. Besides generation capability, we also visually inspect the discriminator's learned representations using t-SNE [247]. Figure 3.2c shows embeddings of the two categories are substantially separated. Figure 3.2d colors each data point by projecting its learned representation into a 2d palette map. From this plot, we see that the learned embedding preserves semantic structure within and across groups. Figure 3.2e shows the gradient of embedding distance approximates the optimal paths between uniform grids and data samples, indicating $\boldsymbol{D}$ learns intrinsic data structure. Figure 3.2f demonstrates the capability of our structural objectives to learn semantic features: when the embedding is updated via $\mathcal{L}$, data that are semantically similar are updated to align in the same directions, whereas data from different clusters diverge.

Method	Parameters (M)	CIFAR-10		CIFAR-100		ImageNet-10	
		SVM	K-M	SVM	K-M	SVM	K-M
Supervised	11.5	95.1	95.1	75.9	73.6	96.4	96.3
Random	11.5	42.9	22.0	18.3	8.9	48.2	28.3
DINO [34]	11.5	89.7	63.9	65.6	36.7	87.8	68.0
NNCLR [70]	11.5	91.7	69.3	69.7	40.4	**91.4**	66.8
SimCLR [40]	11.5	90.6	75.3	65.6	41.3	89.0	65.7
BYOL [87]	11.5	**93.1**	75.0	**70.6**	**42.8**	90.4	67.3
SWAV [33]	11.5	89.1	64.5	65.0	35.2	90.0	61.9
MAE [98]	20.4	82.3	37.0	57.1	17.9	88.4	45.8
DDPM [105]	41.8	91.1	78.0	62.5	36.3	-	-
Ours	11.5	89.8	**80.1**	63.3	38.2	91.2	**75.4**

Table 3.1: *Representation Learning Performance.* We evaluate our trained discriminator by benchmarking its learned representation using linear SVM and K-Means clustering (K-M), reporting average accuracy over 20 runs. Our method, which does not leverage any augmented views, achieves competitive performance with self-supervised approaches across multiple datasets. Compared to denoising autoencoders (shown in the penultimate and antepenultimate rows), our method excels in learning more effective representations while utilizing fewer parameters.

3.5.2 Representation Learning on Real Images

We task the backbone of the discriminator to produce a vector as a data representation and then evaluate its performance on the task of image classification. We compare the results with state-of-the-art contrastive learning approaches under two widely adopted evaluation metrics:

- *Linear Support Vector Machine (SVM)*: We optimize a Linear SVM on top of training feature and report the accuracy on the validation set.
- *K-Means clustering*: We run spherical K-means clustering on the validation set, with K equaling the number of ground-truth categories. We then obtain a prediction on the validation set by solving the optimal assignment problem between the partition produced by clustering and the ground-truth categories. To reduce the randomness in clustering, we repeat this process 20 times and report average performance.

Table 3.1 reports results and provides comparison with current state-of-the-art methods. For datasets with fewer categories, *i.e.,* CIFAR-10 and ImageNet-10, our method significantly out-

Data Aug	Method	CIFAR-10				CIFAR-100			
		KMeans	SVM	LP	1/5 KNN	KMeans	SVM	LP	1/5 KNN
None	SimCLR	14.2	21.8	21.8	14.7/15.1	2.1	4.5	5.4	2.7/2.5
None	Ours	**76.5**	**84.5**	**83.2**	**79.7/82.2**	**22.5**	**52.2**	**51.6**	**37.5 / 37.4**
F	SimCLR	17.5	32.8	32.1	23.3 / 25.5	4.1	10.3	10.9	6.0/5.5
F	Ours	**76.2**	**85.7**	**85.8**	**80.9 / 84.5**	**30.8**	**52.0**	**56.5**	**41.0 / 42.9**
$F + C$	SimCLR	27.8	72.7	72.2	64.9 / 67.2	12.2	35.1	34.3	30.8 / 29.1
$F + C$	Ours	**80.0**	**89.3**	**88.4**	**87.7 / 89.2**	**37.4**	**63.2**	**62.0**	**55.0 / 56.1**
$F + C + J$	SimCLR	78.0	90.7	90.2	88.1 / 89.5	41.7	65.2	65.2	59.2 / 61.4

Table 3.2: *Data Augmentation Dependence.* We compare with SimCLR [40] on sensitivity to various data augmentation schemes. In our system, data augmentation is solely employed to enlarge the training dataset; it is not used for achieving view-consistency objectives. F, C, J denote random horizontal flipping, random image cropping, and color jittering, respectively; None means no augmentation is applied during training. For each augmentation scheme, our method outperforms SimCLR across all evaluation metrics (here, LP denotes linear probing). Moreover, we are able to operate even without data augmentation – a regime in which SimCLR fails.

performs all contrastive learning approaches on the K-means clustering metric. On CIFAR-10, we achieve 80.1% test accuracy, surpassing the best-competing method, SimCLR, which achieves 75.3% test accuracy. On ImageNet-10, our method reaches 75.4% test accuracy surpasses the best-competed method, DINO, with 68.0% test accuracy. When evaluating learned representations using linear SVM, our method reaches 89.8% test accuracy, which exceeds SWAV and DINO, with 89.1% and 89.7% test accuracy respectively, but falls slightly behind BYOL (93.1%) and NNCLR (91.7% test accuracy). On ImageNet10, our method's 91.2% accuracy approaches that of the best method (NNCLR with 91.4%) and exceeds the rest.

CIFAR-100 contains fewer training samples per category and operationalizing instance-wise discriminating objectives is thus favorable over clustering objectives or smoothness regularization. Under such case, our method remains competitive on the linear SVM metric, achieving 63.3% test accuracy, which is very close to DINO, SimCLR, and SWAV, which each have around 65% test accuracy. Using K-Means clustering, our method reaches 38.2% test accuracy, outperforming clustering-based contrastive approaches SWAV (35.2%) and DINO (36.7%).

Our quantitative comparison is also qualitatively confirmed by visualizing embeddings using

Method / Loss	D regularizer	Parameters (M) D	G	IS ↑	FID ↓	K-Means	SVM
StyleGAN2-ADA	Grad Penalty	20.7	19.9	**9.82**	**3.60**	28.96	76.50
BigGAN	Spectral Norm	4.2	4.3	8.22	17.50	29.69	69.31
Hinge Loss	$\mathcal{L}_{\text{reg}}$	11.5	4.9	8.13	18.54	36.41	77.19
Eqn. 3.2 only, D_B	$\mathcal{L}_{\text{reg}}$	11.5	4.9	8.39	17.83	70.76	87.9
Eqn. 3.2 only, JSD	$\mathcal{L}_{\text{reg}}$	11.5	4.9	8.55	16.97	**80.55**	88.32
Full Objectives	Spectral Norm	11.5	4.9	7.23	26.41	55.38	83.9
Full Objectives	$\mathcal{L}_{\text{reg}}$	11.5	4.9	8.73	13.63	80.11	**89.76**

Table 3.4: *Ablation over Loss Function Components on CIFAR-10.* We compare StyleGAN2-ADA [127], BiGAN [28] and a GAN baseline using the standard hinge loss to models using ablated variants of our structural objectives. Dicriminators trained using our objectives significantly outperform these baselines (K-Means, SVM metrics), while our corresponding generators also benefit (IS, FID). Including our finer scale clustering objective (last row) improves both representation and image quality over ablated variants using only our coarse scale objective (rows 4 & 5). The benefit observed when using $\mathcal{L}_{\text{reg}}$ over Spectral norm (final to penultimate row) indicates that preserving model capacity is crucial for effective feature learning.

t-SNE. BYOL, as shown in Figure 3.1, produces isolated and smaller-sized clusters that maintain sufficient space to discern categories under linear transformation. However, those clusters lack sufficient global organization, which is a quality evaluated by the K-Means clustering metric.

Method	NMI	Purity
Self-cond GAN	33.26	11.73
Ours	**72.77**	**81.52**

Table 3.3: *Comparison to Self-conditioned GAN [167] on CIFAR-10.* On normalized mutual information (NMI) and purity metrics, our method outperforms self-conditioned GAN [167], a generative model which clusters discriminator features iteratively in a self-discovering fashion.

In contrast, our approach, as also shown in Figure 3.1, produces smoother embeddings, which are nearly aligned with the ground-truth partition, and consequently yields good K-Means clustering performance. When compared to denoising autoencoders such as DDPM [105] and MAE [98], our model demonstrates superior efficiency by utilizing fewer parameters (11.5M) compared to DDPM (41.8M) and MAE (20.4M). Additionally, our model excels in learning better representations across all evaluated metrics, with the sole exception being a comparison to DDPM on CIFAR-10 (our 89.7% accuracy using SVM *vs.* DDPM's 91.1%).

Method	Network	Linear Probing
GenRep(Tz only)	ResNet-50	55.0*
Ours	ResNet-18	**59.9**

Table 3.5: *ImageNet-100.* Our method outperforms GenRep [115] though we adopt a simpler network architecture and a more direct training pipeline. *For a fair comparison, this result is from Figure 6 of GenRep [115], which does not use data augmentation (Tz only).

3.5.3 Ablation Experiments

Sensitivity to data augmentation. Though we adopt some minimal data augmentation in our experiments, our approach is far less sensitive to data augmentation. However, contrastive self-supervised learning approaches, including SimCLR [40], require a carefully calibrated augmentation scheme to achieve good performance. Table 3.2 highlights this discrepancy. Our method demonstrates a clear advantage over SimCLR across all augmentation regimes, and, unlike SimCLR, can still learn useful features when no augmentation applied.

Comparison to other generative feature learners. GenRep [115] generates images pairs by sampling adjacent features in the latent space of BigBiGAN [63] and then trains an encoder to optimize contrastive objectives. To compare with GenRep, we train our model on ImageNet-100, following most of the our settings for ImageNet-10, except we extend training to 1000 epochs. For fair comparisons, we utilize their *Tz only* version, a setting where no data augmentation is used, and show the results in Table 3.5. Our method outperforms GenRep, though we adopt a simpler network architecture for feature learning.

Self-conditioned GAN [167] clusters the discriminator's features iteratively in a self-discovering fashion; cluster information is fed into the GAN pipeline as conditional input. Though this method produces clustering during training, its objective differs entirely from ours: their motivation is to improve the diversity of image generation, rather than learn representations. Table 3.3 shows that our method outperforms it.

Ablation of system variants. Table 3.4 provides a quantitative comparison of both generator and discriminator performance across baselines as well as ablated and full variants of our system. Our proposed objectives significantly improve generation and representation quality over the hinge loss baseline. We witness further enhancement in image quality when using our extra instance/clustering-wise objective. Performance drops by replacing $\mathcal{L}_{\text{reg}}$ with spectral norm, indicating the effectiveness of our suggested regularization scheme in preserving model capacity. As an additional advantage over spectral norm, we observed better training stability when using our regularization scheme. Note that while StyleGAN2-ADA achieves state-of-the-art generation quality, it both requires adopting a larger network to do so, and still performs worse at feature learning than our system. Appendix B.3 provides a qualitative comparison with examples of generated images.

3.6 Conclusion

Our structural adversarial objectives augment the GAN framework for self-supervised representation learning, shaping the discriminator's output at two levels of granularity: aligning features via mean and variance at coarser scale and grouping features to form local clusters at finer scale. Benchmarks across multiple datasets show that training a GAN with these novel objectives suffices to produce data representations competitive with the state-of-the-art self-supervised learning approaches, while also improving the quality of generated images.

CHAPTER 4
DECIPHERING ‘WHAT’ AND ‘WHERE’ VISUAL PATHWAYS FROM SPECTRAL CLUSTERING OF LAYER-DISTRIBUTED NEURAL REPRESENTATIONS

We present an approach for analyzing grouping information contained within a neural network’s activations, permitting extraction of spatial layout and semantic segmentation from the behavior of large pre-trained vision models. Unlike prior work, our method conducts a holistic analysis of a network’s activation state, leveraging features from all layers and obviating the need to guess which part of the model contains relevant information. Motivated by classic spectral clustering, we formulate this analysis in terms of an optimization objective involving a set of affinity matrices, each formed by comparing features within a different layer. Solving this optimization problem using gradient descent allows our technique to scale from single images to dataset-level analysis, including, in the latter, both intra- and inter-image relationships. Analyzing a pre-trained generative transformer provides insight into the computational strategy learned by such models. Equating affinity with key-query similarity across attention layers yields eigenvectors encoding scene spatial layout, whereas defining affinity by value vector similarity yields eigenvectors encoding object identity. This result suggests that key and query vectors coordinate attentional information flow according to spatial proximity (a ‘where’ pathway), while value vectors refine a semantic category representation (a ‘what’ pathway).

4.1 Introduction

An explosion in self-supervised learning techniques, including adversarial [85, 126, 122], contrastive [266, 97, 44, 40], reconstructive [136, 246], and denoising [224, 105] approaches, combined with the focus on training large-scale foundation models [27] on vast collections of image data has produced deep neural networks exhibiting dramatic new capabilities. Recent examples of

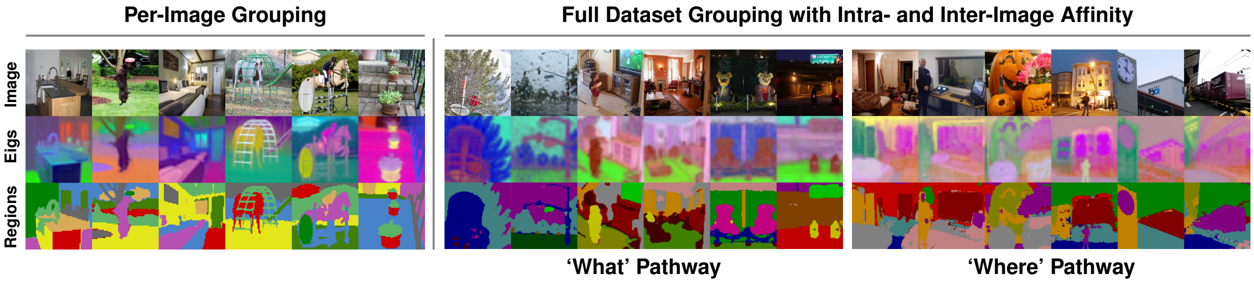


Figure 4.1: Our novel optimization procedure, resembling spectral clustering, leverages features throughout layers of a pre-trained model to extract dense structural representations of images. Shown are results of applying our method to Stable Diffusion [204]. ***Left:*** Analyzing internal feature affinity for a single input image yields region grouping. ***Right:*** Extending the affinity graph across images yields coherent dataset-level segmentation and reveals 'what' (object identity) and 'where' (spatial location) pathways, depending on the feature source.

such models include CLIP [195], DINO [34], MAE [98], and Stable Diffusion [204]. As training is no longer primarily driven by annotated data, there is a critical need to understand what these models have learned, provide interpretable insight into how they work, and develop techniques for porting their learned representations for use in accomplishing additional tasks.

However, interpretable analysis of neural networks is challenging. Procedures such as guided backpropagation [230] or Grad-CAM [217] assist with interpretability with respect to particular labels, but are limited in scope. Others propose heuristics for extracting information relevant to particular downstream tasks, or analyze specific features in models [62, 155, 164, 294, 238, 46, 99, 37, 38, 11]. The distributed nature of both the information encoded within deep networks [234] and their computational structure frustrates the development of general-purpose techniques.

It is similarly unclear how best to repurpose pre-trained models toward downstream tasks. Task-specific heuristics, fine-tuning on labeled data, prompt engineering (if applicable), or clustering frozen feature representations might all be viable options. Yet, an element of art remains in choosing which features to extract or which layers to fine-tune.

We introduce a new analysis approach that provides insight into model function and directly extracts significant visual information about image segmentation, as shown in Figure 4.1, with neither a-priori knowledge of, nor hyperparameter search over, where such information is stored in the network. We accomplish this through an analysis that couples the entire activation state of the network, from shallow to deep layers, into a global spectral clustering objective. Solving this clustering problem not only yields new feature representations (in the form of eigenvector embeddings) directly relevant to downstream segmentation tasks, but also, as Figure 4.2 illustrates, provides insight into the inner workings of vision models. Our contributions include:

- A new approach, inspired by spectral clustering, for wholistic analysis of deep neural network activations.
- Improved quality of extracted regions across models, compared to variants analyzing single layers.
- An efficient gradient-based optimization framework that enables our approach to scale to joint analysis of network behavior across an entire dataset simultaneously.
- Unsupervised semantic segmentation results on par with STEGO [92], but extracted from a pretrained generative model rather than a contrastive backbone.
- Insight into the computational strategy learned by large-scale vision models: internal features are partitioned into 'what' and 'where' pathways, which separately maintain semantic and spatial information.

4.2 Related Work

Image segmentation. Segmentation, as a generic grouping process, has historically been regarded as an important intermediate task in computer vision. Significant efforts focus on building object-agnostic methods for partitioning an image into coherent regions or, equivalently, their dual representation as contours [30, 4, 202, 61, 19, 141], with standard benchmarks [175] driving progress. Semantic and instance segmentation, which aim to extract image regions corresponding to specific category labels or object instances, have undergone parallel development, driven by benchmark datasets such as PASCAL [71] and COCO [163]. Notable modern supervised methods utilize CNN [96] or Transformer [32] architectures trained in an end-to-end fashion. Particularly relevant is recent work demonstrating the ability of models to learn to segment with relatively few labels [11, 299]. Spectral clustering, as a method of approximating the solution of a graph partitioning objective [220], has appeared as a core algorithmic component across a variety of segmentation systems [220, 278, 277, 4, 173, 141, 174, 239].

Segmentation without labels. Recent methods, such as DINO [34], learn intra-image and inter-image correspondences between pixels without the need for dense labels. STEGO [92] and PiCIE [49] propose to cluster pixel-wise features of a self-supervised backbone, showing impressive performance on semantic segmentation with no labels at all. LSeg [154] and GroupViT [268] modify CLIP [195] to enable zero-shot open-vocabulary semantic segmentation.

Another class of methods builds entirely on top of existing models, with no additional training [62, 155, 164, 294]. Recent attempts at instance segmentation [254, 257] yield impressive results through heuristic decoding strategies based on the structure of a particular model's features (*e.g.,* the final layer of DINO [34]). Other work, based on Stable Diffusion [204], finds unsupervised dense correspondences using the text embedding space as a shared anchor [99] or through careful choice of features [238].

Interpretability. Grad-CAM [217], layer-wise relevance propagation [180], and guided backpropagation [230] provide heuristics to visualize the responsibility of different input spatial regions for predictions of a deep neural network. Other approaches visualize attention matrices to find salient input regions for NLP [50, 142] and vision [270, 37, 38] tasks. Chen *et al.* [46] find evidence of depth information inside Stable Diffusion. Yet, visualizing and interpreting neural network behavior remains a challenging problem due to the distributed nature of the representations they learn [234].

Spectral clustering of neural features. TokenCut [257], MaskCut [255], and DSM [177] define affinity graphs using final features of a pre-trained DINO [34] model, and use spectral clustering to segment the original image. We take inspiration from these approaches and utilize them as baselines for experimental comparison. Our methodology differs in being global and accounting for features throughout the network, rather than restricted to one layer.

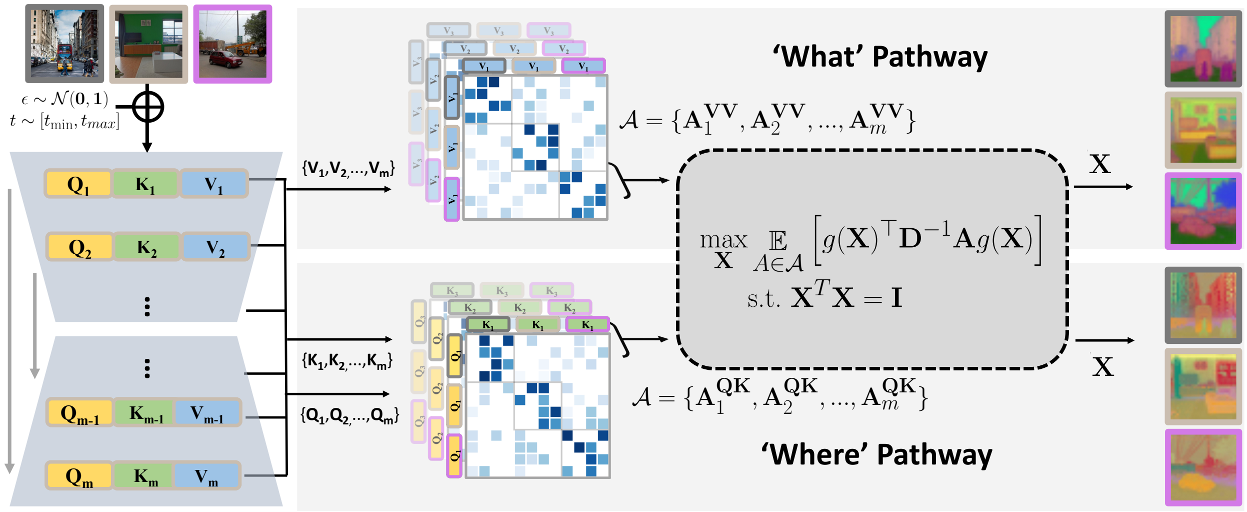
Neuroscience perspectives on visual processing streams. Trevarthen [244] and Schneider [215] propose the concept of separate visual processing pathways in the brain for localization ('where') and discrimination ('what'). Mishkin *et al.* [178] review evidence for this specialization of processing in the monkey, while subsequent work examines specialized pathways in terms of perception and action [84], as well as spatial memory and navigation [143]. While these ideas motivate our investigation into information stored in the key, query, and value vectors distributed throughout a Transformer architecture, the question of relevance (if any) to biological vision systems is beyond our scope.

4.3 Method

Our method closely resembles spectral clustering applied simultaneously across attention layers within a given neural network. The following sections detail our full method and discuss different graph construction choices for spectral clustering, which respectively allow us to extract different kinds of information from source models.

4.3.1 Spectral Clustering with Distributed Features

Spectral clustering formulates the data grouping problem from the view of graph partitioning. It uses the eigenvectors of the normalized Laplacian matrix to partition the data into balanced subgraphs with minimal cost of breaking edges [220]. Specifically, with a symmetric affinity matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$, where N denotes the total number of data points and entries $\boldsymbol{A}_{ij} \geq 0$ measure the similarity between data samples with indices i and j, we can embed the data into a lower dimensional representation $\boldsymbol{X} \in \mathbb{R}^{N \times C}$, where C denotes the number of feature channels



$\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$
$t \sim [t_{\min}, t_{max}]$
Q_1 K_1 V_1
Q_2 K_2 V_2
Q_{m-1} K_{m-1} V_{m-1}
Q_m K_m V_m
$\{V_1, V_2, \ldots, V_m\}$
$\{K_1, K_2, \ldots, K_m\}$
$\{Q_1, Q_2, \ldots, Q_m\}$
'What' Pathway
$\mathcal{A} = \{\mathbf{A}_1^{\mathbf{VV}}, \mathbf{A}_2^{\mathbf{VV}}, \ldots, \mathbf{A}_m^{\mathbf{VV}}\}$
$\max_{\mathbf{X}} \mathbb{E}_{A \in \mathcal{A}} \left[g(\mathbf{X})^\top \mathbf{D}^{-1} \mathbf{A} g(\mathbf{X}) \right]$
s.t. $\mathbf{X}^T \mathbf{X} = \mathbf{I}$
$\mathbf{X}$
$\mathbf{X}$
$\mathcal{A} = \{\mathbf{A}_1^{\mathbf{QK}}, \mathbf{A}_2^{\mathbf{QK}}, \ldots, \mathbf{A}_m^{\mathbf{QK}}\}$
'Where' Pathway

Figure 4.2: **Spectral clustering of layer-distributed representations.** For each input image, we collect key, query, and value feature vectors from attention layers across network depth (and, for diffusion models, time). Intra- and inter-image value-value (top) and key-query (bottom) similarity define a collection of affinity matrices indexed by layer (and time). We solve for pseudo-eigenvectors $\boldsymbol{X}$ which, when scaled to the spatial resolution of each layer via $g(\cdot)$, best satisfy an average of per-layer spectral partitioning criteria. The leading eigenvector from value-value affinity reveals semantic category *(top)*, while that from key-query affinity reveals spatial layout *(bottom)*.

and $C \ll N$, as the solution of the following generalized eigenproblem:

$$(\boldsymbol{D} - \boldsymbol{A})\boldsymbol{X} = \lambda \boldsymbol{D}\boldsymbol{X}. \tag{4.1}$$

$\boldsymbol{D}$ is the diagonal degree matrix of $\boldsymbol{A}$ with diagonal entries $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{A}_{ij}$, and $\boldsymbol{X}, \lambda$ are eigenvectors and eigenvalues respectively. We can then produce a discrete partition from $\boldsymbol{X}$ through K-Means clustering.

Though spectral clustering is a powerful tool for data analysis, its performance is highly dependent on the choice of affinity matrix. Recent works [257, 255, 177] apply spectral clustering on an affinity matrix constructed from features in the last layer of DINO [34], yielding strong performance in segmentation tasks. However, the choice of graph may not be clear when the desired information is distributed across the layers of a neural network, or noise levels in diffusion models [11]. Therefore, we extend Eqn. 4.1 to allow for constructing $\boldsymbol{A}$ using multiple sources of information. A classic approach to solve Eqn. 4.1 with a set of affinity matrices, $\mathcal{A} = \{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_m\}$, is the constrained spectral clustering problem [54]. It constructs a block diagonal affinity matrix from this set:

$$\boldsymbol{A}_{\mathcal{A}} = \begin{bmatrix} \boldsymbol{A}_1 & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{A}_m \end{bmatrix} \tag{4.2}$$

and imposes additional cross-scale consistency constraints. However, the size of this matrix, and the computational expense of solving the resulting eigenproblem, can become intractable with increasing $|\mathcal{A}|$. Instead of solving the original eigenproblem in Eqn. 4.1, we solve an approximation:

$$\begin{aligned} &\max_{\boldsymbol{X}} \mathop{\mathbb{E}}_{\boldsymbol{A} \in \mathcal{A}} \left[g(\boldsymbol{X})^\top \boldsymbol{D}_{\boldsymbol{A}}^{-1} \boldsymbol{A} g(\boldsymbol{X}) \right], \\ &\text{s.t. } \boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I} \end{aligned} \tag{4.3}$$

where $g(\cdot)$ corresponds to the resampling function that bilinearly interpolates the spatial resolution of $\boldsymbol{X}$ to match the size of $\boldsymbol{A}$, allowing affinity matrices to be constructed from feature maps with varying resolutions. In Eqn. 4.3, we follow Meila and Shi [176] to solve the spectral clustering from the random walk perspective, since the random walk matrix and the attention matrix have the same format and the same eigenvectors. This objective encodes a Rayleigh quotient optimization simultaneously across affinities in $\mathcal{A}$, which avoids the intractable exact solution and can naturally scale with increasing $|\mathcal{A}|$.

Notice that $\boldsymbol{D}_{\boldsymbol{A}}^{-1}\boldsymbol{A}$ is a random walk matrix with maximum eigenvalue of 1. For numerical stability, we impose a constraint to ensure that the maximum value of the objective does not exceed 1. In addition, we replace the strict orthogonality requirement with a soft Frobenius regularization term whose coefficient is 1. Consequently, our final optimization objective is:

$$\min_{\boldsymbol{X}} \mathop{\mathbb{E}}_{\boldsymbol{A}\in\mathcal{A}} |g(\boldsymbol{X})^\top \boldsymbol{D}_{\boldsymbol{A}}^{-1} \boldsymbol{A} g(\boldsymbol{X}) - 1| + \|\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{I}\|_F. \tag{4.4}$$

We parameterize $\boldsymbol{X}$ as a learnable feature map and solve for it using gradient-based optimization. In the following subsections, we discuss the choice of affinity set $\mathcal{A}$ and how that choice affects the information we extract.

4.3.2 Per-Image Analysis

Attention layers in vision models naturally consider patch-wise relationships when computing the attention matrix. We can use this matrix as an affinity graph for spectral clustering, which allows investigating how a model groups regions in an image internally, without imposing outside heuristics. For the Vision Transformer [66] and U-Net [205] variants that include a total of m attention blocks, we build an affinity set $\mathcal{A} = \{\boldsymbol{A}_l^{\boldsymbol{QK}}\}_{l=1}^m$ across layers, where $\boldsymbol{A}_l^{\boldsymbol{QK}}$ is the pre-softmax self-attention matrix [248] at layer l.

$$\boldsymbol{A}_l^{\boldsymbol{QK}} = \exp\left(\frac{\boldsymbol{Q}_l \boldsymbol{K}_l^\top}{\sqrt{d_l}}\right) \in \mathbb{R}^{N\times N}, \tag{4.5}$$

where $\boldsymbol{Q}_l, \boldsymbol{K}_l \in \mathbb{R}^{N\times d_l}$ are the query and key matrices for that layer respectively, and d_l is the embedding dimension.

4.3.3 Full-Dataset Extension

We can extend the self-attention operation in a single image to affinity matrix construction across different images. This allows probing how models relate different regions across different images using their internal computational structure. Specifically, we construct graphs similar to single-image self-attention matrices by computing normalized pairwise dot products between queries at every position in one image, and keys at every position in another. Scaling to large datasets, we extract one set of features $\boldsymbol{X}_i$ for each image with index i in the dataset. To do this, we optimize a mini-batch of features:

$$\boldsymbol{X}_{\text{batch}} = \begin{bmatrix} \boldsymbol{X}_j \\ \vdots \\ \boldsymbol{X}_k \end{bmatrix} \in \mathbb{R}^{(N \cdot B) \times C}, \tag{4.6}$$

and construct graphs over that mini-batch:

$$\boldsymbol{A}_l^{\boldsymbol{QK}} = \begin{bmatrix} \widehat{\boldsymbol{Q}}_{j,l} \\ \vdots \\ \widehat{\boldsymbol{Q}}_{k,l} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{K}}_{j,l}^{\top} \dots \widehat{\boldsymbol{K}}_{k,l}^{\top} \end{bmatrix} \in \mathbb{R}^{(N \cdot B) \times (N \cdot B)}, \tag{4.7}$$

where $\widehat{\boldsymbol{Q}}_{j,l}, \widehat{\boldsymbol{K}}_{j,l} \in \mathbb{R}^{N \times d_l}$ represent the queries and keys for image j at layer l normalized to unit-norm, and there are B images in a mini-batch. We normalize vectors as calibrating magnitudes across images is not trivial.

Though we limit the graph to a mini-batch, it is still prohibitively expensive to store and optimize over. Thus, we sparsify the graph by only keeping the top c_{intra} intra-image connections and the top c_{inter} inter-image connections for each location. In addition, we set all values below a threshold to 0. To investigate what kind of information models mix across spatial locations, we consider a similar affinity set $\mathcal{A} = \{\boldsymbol{A}_l^{\boldsymbol{VV}}\}_{l=1}^m$ built from the value matrices $\widehat{\boldsymbol{V}}_{i,l}$.

With these approximate layer-wise graphs, we optimize the objective in Eqn. 4.4 a small number of steps per mini-batch, then sample a new mini-batch of images and continue. Finally, this process discovers a consistent set of dense features for a dataset. A visualization of the entire method can be found in Figure 4.2.

4.3.4 Recovering Orthogonal Representations

Eqn. 4.4 suggests an approximate formulation of the spectral clustering problem. While this results in a structured $\boldsymbol{X}$, it fails to enforce an orthogonal representation capable of separating distinct features into channels. To overcome this, we orthogonalize $\boldsymbol{X}$ by finding the eigenvectors $\boldsymbol{U}$ of a small matrix $\boldsymbol{X}^\top \boldsymbol{X} \in \mathbb{R}^{C \times C}$. This is similar to the reorthogonalization step in approximate eigensolvers; *e.g.,* lines 36-38 of Algorithm 2 in Maire and Yu [172]. The final representation is given by:

$$\boldsymbol{X}_{\text{ortho}} = \boldsymbol{X}\boldsymbol{U}. \tag{4.8}$$

After extracting these final features, we create hard assignments using K-Means clustering.

4.4 Experiments

Leveraging our method, we investigate how models group image regions internally. In Section 4.4.1 we see how models associate locations within an image. Section 4.4.2 examines the same behavior across images and discovers a spatial/semantic split depending on the choice of internal features used for grouping. We evaluate this phenomenon quantitatively, deriving a high quality training-free unsupervised semantic segmentation from Stable Diffusion [204] in Section 4.4.2.2, as well as providing stronger evidence for spatial information pathways in Section 4.4.2.3.

4.4.1 Per-Image Region Extraction

To show how models partition images spatially, we extract dense eigenvectors for individual images and cluster these features into hard segmentations, as detailed in Section 4.3.2.

Experimental Setup. For all models, during optimization we consider all heads of all self-attention layers to be independent graphs. In the case of Stable Diffusion, this is 16 self-attention layers with 8 attention heads, thus $|\mathcal{A}| = 16 \times 8 = 128$ per forward pass. Specific to Stable Diffusion, in each iteration we add noise to the input image by randomly sampling noise timestep

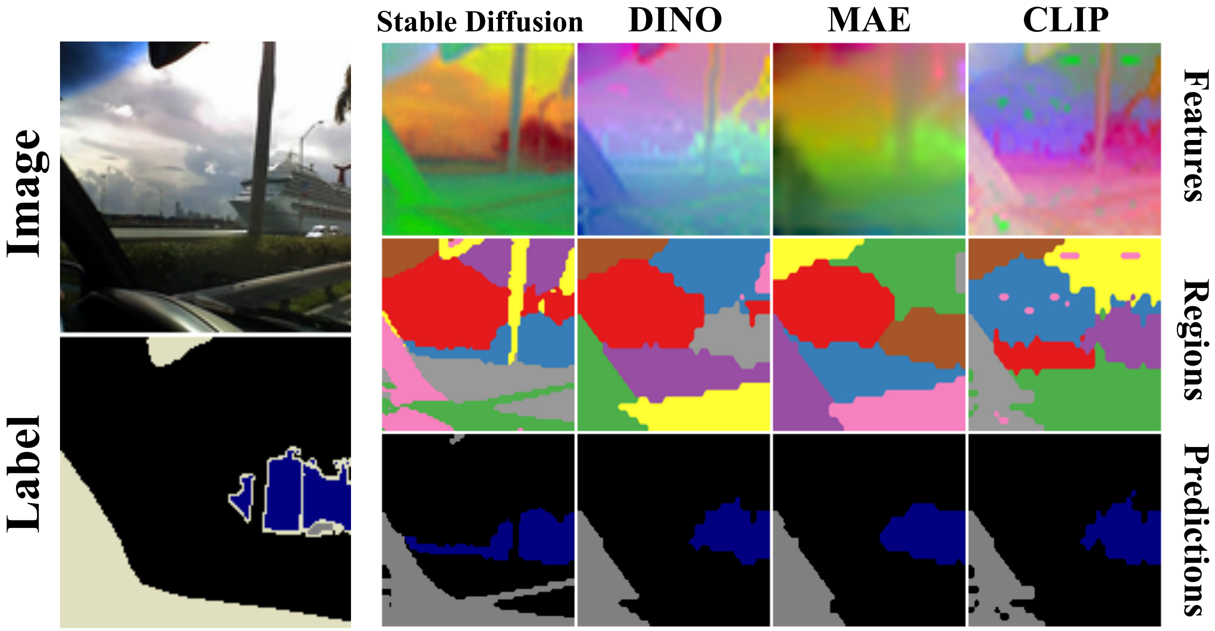


Figure 4.3: **Features extracted from different models on PASCAL VOC [71].** Across models we extract meaningful regions, even for models like Stable Diffusion [204], CLIP [195] or MAE [98] whose training is not well-aligned with segmentation.

Config	**All**	Enc	Mid	Dec	32x32	64x64
Layer Index	1-16	1-4	5 - 10	11-16	3-4 11-13	1-2 14-16
mIoU	0.75	0.66	0.76	**0.80**	0.75	0.70

t_{max}	250	**500**	750	999
mIoU	**0.77**	0.75	0.74	0.69

Table 4.1: Ablation of layer index and maximum noise level of the diffusion model on the PASCAL VOC dataset [71]. We find that using only decoder layers and middle noise yields the best results.

$t \in \mathcal{U}[0, 500)$. For all models, we construct feature map $\boldsymbol{X}$ with spatial resolution matching the finest attention layer resolution and set $C = 10$. We initialize $\boldsymbol{X}$ from a normal distribution and solve the optimization problem with Adam for ~ 2000 iterations with learning rate 1e-3.

To produce discrete regions, we run K-Means clustering by sweeping K from 2 to 10 and use silhouette score [206] to select the best value. To speed up extraction in Stable Diffusion, we cache attention matrices into a buffer for reuse with a 90% chance, bringing the per-image runtime from 154 to 67 seconds. For more implementation details, please refer to Appendix C.2. To

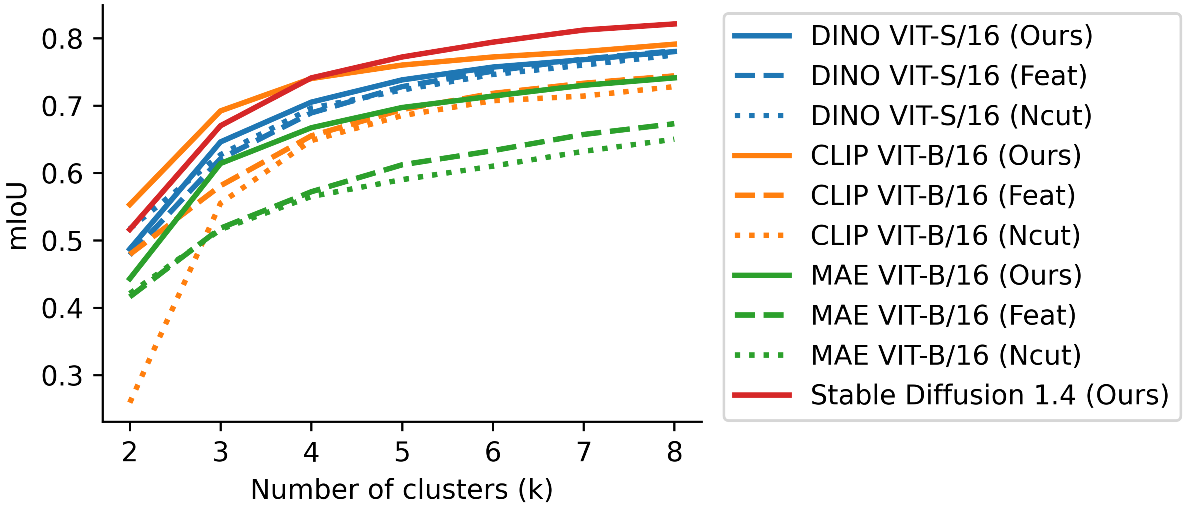


Figure 4.4: **Oracle-based semantic segmentation performance with varying region count.** Across models and number of clusters (regions) returned by K-Means, our method (Ours + K-Means) yields better agreement (in mIoU) with ground-truth than running Normalized Cuts (Ncut + K-Means), or directly applying K-Means on the final output features of the model (K-Means). We observe an even more significant improvement when applying our method to MAE and CLIP, which do not produce discriminative features.

provide a measure for comparison, we extract multiple regions according to two related methods: Normalized Cut [220] and MaskCut [255]. Both of these methods require a single affinity matrix, the choice of which we ablate in Appendix C.2.

In Figure 4.3, we show that features and regions extracted from different models are quite structured, aligning well with object boundaries. We quantify region quality by measuring their oracle overlap with semantic segmentation labels. This gives a sense as to how well attention layers inside models decompose images along semantic axes. We perform this analysis on PASCAL VOC [71], which has 20 foreground classes and 1449 validation images. We score results with the metric of mean intersection over union (mIoU) between regions and labels. Each region is assigned to the ground-truth label it overlaps with the most.

Results and Analysis. Table C.1 presents results demonstrating that our approach consistently outperforms all methods to which we compare, across various backbone models. For DINO, we show that directly clustering the final layer features using K-Means yields decent performance. This is likely due to the discriminative nature of DINO's final representation, which makes a straightforward decoding strategy sufficient for generating satisfactory regions. However, direct clustering fares much worse on other models with different training objectives.

Additionally, we observe that Normalized Cut [220] (Ncut) is highly sensitive to the underlying graph, and its performance deteriorates significantly when switching from the graph of final features to the final attention matrix. A related approach, MaskCut [255], solves Ncut on a binarized graph to extract foreground objects. However, this operation results in the loss of finer-grained information, which is crucial for segmentation tasks. In contrast, our method is less sensitive to the quality of a single graph because we simultaneously perform spectral clustering over a set of affinity matrices. When comparing our method on models that are not trained to produce discriminative features as their final output, such as MAE and CLIP, we observe an even more substantial improvement.

In Table 4.1, we provide ablation studies on the choices of layer for feature extraction and maximum noise level. Stable diffusion has 16 attention layers with resolutions from 64x64 to 8x8. Our default is All (1-16), $t_{\max}$ = 500. We conduct experiments on a per-image region extraction setting with 200 images from the PASCAL VOC validation set. Although our main experiment utilizes features from all layers, making minimal assumptions about layer-wise feature distribution, we find that using only decoder layers and a middle noise level yields better results.

To further evaluate the region quality irrespective of decoding choices, Figure 4.4 shows the mIoU with varying choice of K. We see that the high quality of regions persists across choices, even when compared with baselines.

Our per-image regions can find broad applicability in a variety of segmentation tasks. For first-step proof-of-concepts, see Appendices C.4.1 and C.4.2.

4.4.2 Full-Dataset Region Extraction

Our method can effectively extract regions within images. Can it examine relationships across images? To probe different kinds of encoded information, we take the best model of the previous section, Stable Diffusion, as a case-study. We compare the query/key (Q-K) dataset-level graph with the value/value (V-V) dataset-level graph, as described in Section 4.3.3. Results show a surprisingly structured split, where Q-K encodes spatial information and V-V encodes semantic information, which we can use for tasks like unsupervised semantic segmentation.

Experimental Setup. For constructing graphs, we follow the method in Section 4.3.3. For efficiency, we concatenate features at each head into a single vector instead of considering heads independently. We select one attention block in the middle block and the first 6 attention blocks in the up-sampling blocks, resulting in a total of 7 attention matrices. We choose channel number $C = 50$, cross-image connections $c_{\text{inter}} = c_{\text{intra}} = 10$, noise level $t \in \mathcal{U}[20, 300)$, and optimize using Adam [135] with a learning rate of 1e-2, and a batch size of 160 images over 4 GPUs for 2100 iterations. When clustering, we choose K to be the number of labels of the relevant task. More details are in Appendix C.3.

4.4.2.1 Qualitative Analysis

We show qualitative results for both Q-K and V-V graphs on COCO [163] in Figure 4.5 and Cityscapes [53] in Figure 4.6.

Across datasets, we observe that the Q-K graph appears to encode spatial relationships. On Cityscapes, which has a clear spatial layout at the scene level, the learned eigenvectors effectively separate buildings, cars, people, and trees into left/right subgroups. For the more complex dataset COCO, which lacks fixed spatial patterns at the scene level, the eigenvectors uncover spatial correlations first in terms of ground, subject, and background, and then part-like correlations within objects from top-to-bottom.

By contrast, features from the V-V graph group objects semantically. In COCO, we observe that eigenvectors encode semantic structure hierarchically: the first set of eigenvectors focuses on distinguishing scene-level semantics (*e.g.,* ground, sky, trees) while overlooking differentiating foreground objects. The next set of eigenvectors groups foreground objects like people, animals, and vehicles. In Cityscapes, the initial set captures broad scene-level semantics, including trees, houses, and the egocentric vehicle, and can differentiate between road and sidewalk. The following set groups cars, people, and road markings. More examples are available in Appendix C.1.

The qualitative differences between eigenvectors stemming from the Q-K and V-V graphs suggest a clear split in the way the model processes information from images. In attention layers, queries and keys are used to form patch-wise relationships, which then modulate the values propagated to the next stage. It appears that the model learns to split representation into spatial and semantic branches as a convenient solution for taking advantage of this computational structure. This is perhaps surprising, as the projection from features to queries or keys or values need not split information in such a clean fashion. The discrepancy in the behavior between features from these different graphs motivates us to further evaluate their performance in separate semantic and spatial benchmarks.

4.4.2.2 Quantitative Analysis for 'What' Pathway

To quantify semantic segmentation ability in the V-V graph, we evaluate our extracted segmentation on two common unsupervised semantic segmentation tasks: COCO-Stuff [163, 29] and Cityscapes [53]. We follow the preprocessing protocol as adopted in PiCIE [49] and STEGO [92]. We optimize $\boldsymbol{X}$ on the validation set, where images are first resized so the minor edge is 320px and then cropped in the center to produce square images. We choose $K = 27$, the number of ground-truth categories in both datasets, for K-Means over $\boldsymbol{X}$, and then use greedy matching to align the cluster assignments with the ground truth. We report results with mIoU and compare to other methods in Table 4.2, and examine feature choice and decoding protocol in Table 4.3. More

Method	**Results (mIOU)**	
	COCO-Stuff-27	**Cityscapes**
MoCo v2 [44]	4.4	-
IIC [117]	6.7	6.1
DSM [177] (ViT-B/8)	8.9	-
Modified DC [49]	9.8	7.4
PiCIE [49]	13.8	12.3
PiCIE+H [49]	14.4	-
ACSeg [155]	16.4	-
HP [219] (ViT-S/8)	24.6	18.4
STEGO [92] (ViT-B/8)	26.8	18.2
STEGO [92]+CRF (ViT-B/8)	**28.2**	**21.0**
Ours (V-V graph)	25.4	16.2
Ours (V-V graph)+CRF	27.1	16.9

Table 4.2: **Unsupervised semantic segmentation results on COCO-Stuff-27 and Cityscapes.** We observe that the V-V graph features outperform those of prior works and achieve competitive performance compared to the strong STEGO method, which utilizes discriminative DINO [34] features and a complex two-stage global nearest-neighbor strategy. Conversely, our method employs representations from a generative model and collects neighbors solely from the minibatch, a simpler and more scalable approach.

details are in Appendix C.3.

In Table 4.2, our method significantly outperforms many other methods and is comparable to STEGO [92]. This is quite surprising, as STEGO [92] adopts a sophisticated two-stage dataset-wise nearest-neighbor searching procedure, while our method only considers connections within the mini-batch, a strategy with noisy signal but with better scalability. STEGO [92] also benefits from the discriminative representations of DINO [34], while our backbone, Stable Diffusion [204], is generative.

Table 4.3 reports results on directly clustering the most semantic representations of Stable Diffusion [204], which are the features of the 2nd upsampling block with timestep $t = 250$ [238]. In this comparison, DINO [34] features are better than Stable Diffusion, likely due to their discriminative properties, but our method greatly narrows the gap.

Method	COCO-Stuff-27		Cityscapes	
	Greedy	Hungarian	Greedy	Hungarian
K-Means (SD [204])	9.2	8.6	12.4	8.1
K-Means (DINO [34])	13.7	13.0	13.3	8.7
K-Means (STEGO [92])	26.6	24.0	15.8	14.9
STEGO [92]	**27.0**	**26.5**	**16.6**	**18.2**
Ours (Q-K graph)	12.4	10.9	10.91	9.7
Ours (V-V graph)	25.4	23.2	16.2	11.4

Table 4.3: **Ablations for unsupervised semantic segmentation.** We test multiple sources of features for clustering on the validation set only, and vary the decoding pipeline for evaluation. With greedy decoding, our features are comparable, but with Hungarian matching STEGO is stronger. We also see that the Q-K graph encodes far less semantic information than the V-V graph, supporting a semantic/spatial decomposition. The discrepancy between K-Means (STEGO) and STEGO numbers here is due to restricting clustering to the validation set.

4.4.2.3 Quantitative Analysis for ‘Where’ Pathway

Here, we design two experiments to quantify the positional information in the Q-K graph for the ‘where’ pathway. Our first experiment aims to measure the amount of positional information contained within the features. In this setting, we train a linear head on top of the features and attempt to regress the corresponding grid position of each pixel/patch. We call this task “coordinate regression".

The second experiment aims to evaluate whether this spatial information is present at the semantic level. In this case, we benchmark on an unsupervised semantic segmentation task by further partitioning the semantic annotations into left and right subgroups. For this purpose, we process the ground-truth annotations by first identifying disconnected regions for each category of segmentation annotation and then scoring each region based on its pixel distance to the image’s left/right border. We filter out smaller regions with pixel counts less than 50 and ambiguous regions located close to the image center. We call this task “spatial semantic segmentation” and showcase the original and processed semantic segmentation maps in Figure 4.7. We follow the exact same evaluation protocol as used in the experiment for the ‘what’ pathway.

The results of both experiments are presented in Table 4.4, where we compare with STEGO features, DINO final-layer features, and ground-truth semantic segmentation labels. In both exper-

Method	Coordinate Regression (MSE) ↓	Spatial Semantic Segmentation (mIOU) ↑
DINO	**3.2**	6.0
GT semantic label	72.0	-
STEGO	42.4	6.9
Ours (V-V graph)	43.1	5.1
Ours (Q-K graph)	**19.5**	**10.1**

Table 4.4: **Results for evaluating 'where' pathway on spatial structures**. Ours (Q-K graph) outperforms STEGO and ours (V-V graph) in both benchmarks suggesting the Q-K graph contains richer spatial information at object levels. Though DINO can trivially recover the spatial coordinates through positional embeddings, it fails to leverage that information for segmentation.

iments, our approach with the Q-K graph outperforms both STEGO and the variants with the V-V graph.

Results on coordinate regression suggest that $\boldsymbol{X}$ from the Q-K graph contains rich spatial information for processing the 'where' pathway. However, both STEGO and the V-V graph group pixels only by semantic similarity and remove spatial information from the final representation. DINO performs well on regression likely due to position embeddings.

We further verify the spatial information content of the Q-K graph by examining the results of spatial semantic segmentation. We see that these features are strongest in this task. Compared to results on unsupervised semantic segmentation (Table 4.2), the strong performance of features from the V-V graph and STEGO deteriorates due to failure to reason about spatial structure. DINO features also fail in this task, likely as spatial information is not as strong a signal as semantics for discriminating between images. These results, along with those in Section 4.4.2.2, show that our approach can scale efficiently to extract both spatial and semantic relationships across images.

4.5 Discussion

We present an approach for extracting information from a neural network's activations. Unlike prior work, our method examines the whole of a network, without needing to guess which part of the model contains relevant features. Our approach resembles classic spectral clustering, but gains scalability to dataset-level analysis by approximating a solution using gradient-based optimization.

Deployed as a mechanism for extracting image segmentation from large pre-trained models, we observe robust performance in producing regions from a wide variety of source models, including high quality semantic segmentations obtained from a Stable Diffusion model. Deployed as an analysis tool, we gain new insight into how vision models with attention layers utilize key, query, and value vectors to coordinate the flow of spatial and semantic information, and disentangle ‘what’ and ‘where’ pathways within these deep networks.

Our approach could be the first example in a new class of optimization-centric techniques for peering into the inner workings of deep networks. Future research could repurpose other computationally intensive, but scalable, classic machine learning tools to the analysis of network behavior.

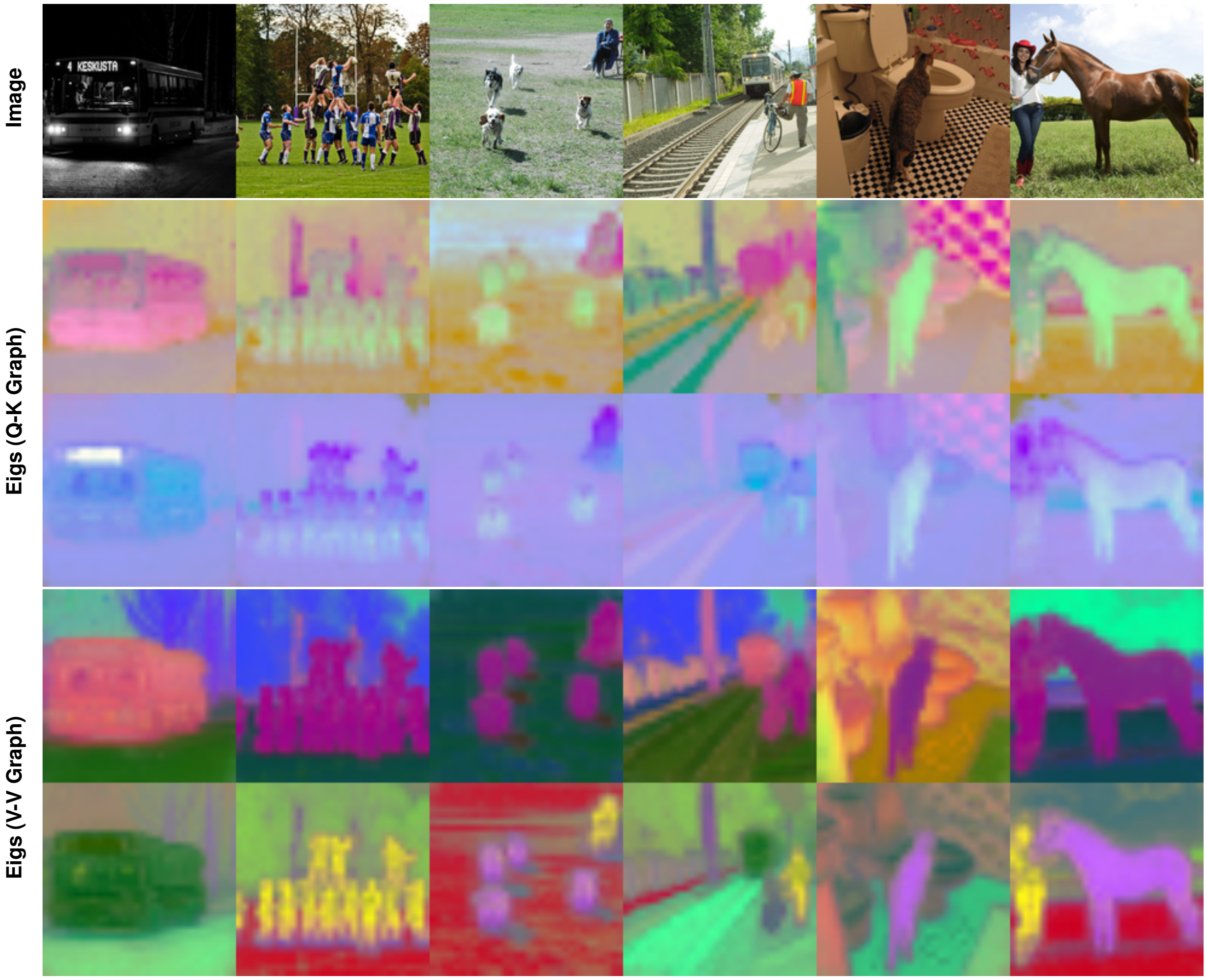


Figure 4.5: **Extracted eigenvectors on COCO for both graph choices.** We visualize selected components of $\boldsymbol{X}_{\text{ortho}}$, sorted by decreasing eigenvalue. Three eigenvectors at a time are rendered as RGB images. In the Q-K case, the first set of eigenvectors describes general scene spatial layout in terms of ground, subject, background, and sky. The second finds top-to-bottom part separation within objects. In the V-V case, the first set of eigenvectors partitions the image into coarse semantics like trees, ground, and sky, while the second set recognizes finer-grained categories and groups individual objects like people, animals, and vehicles.

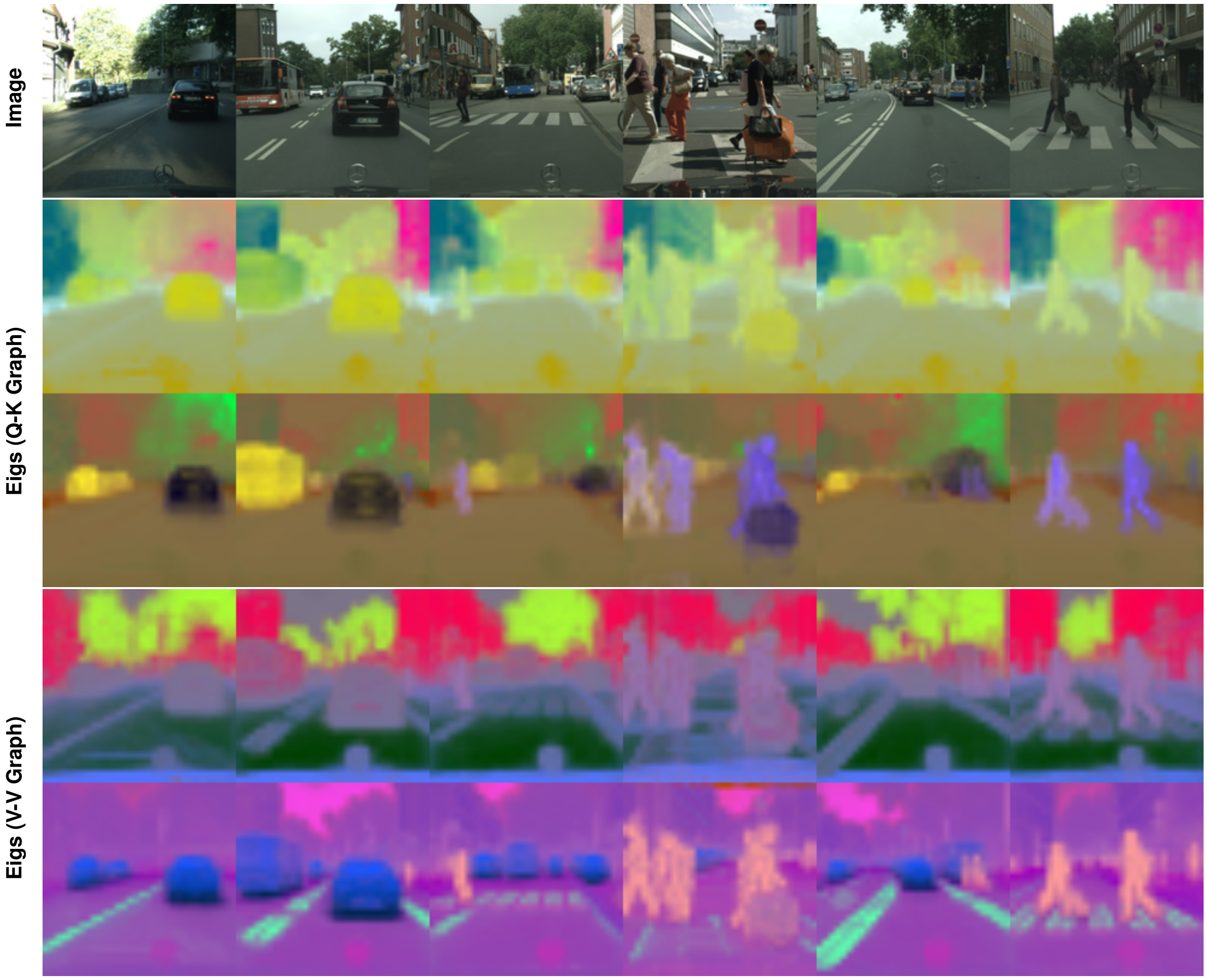
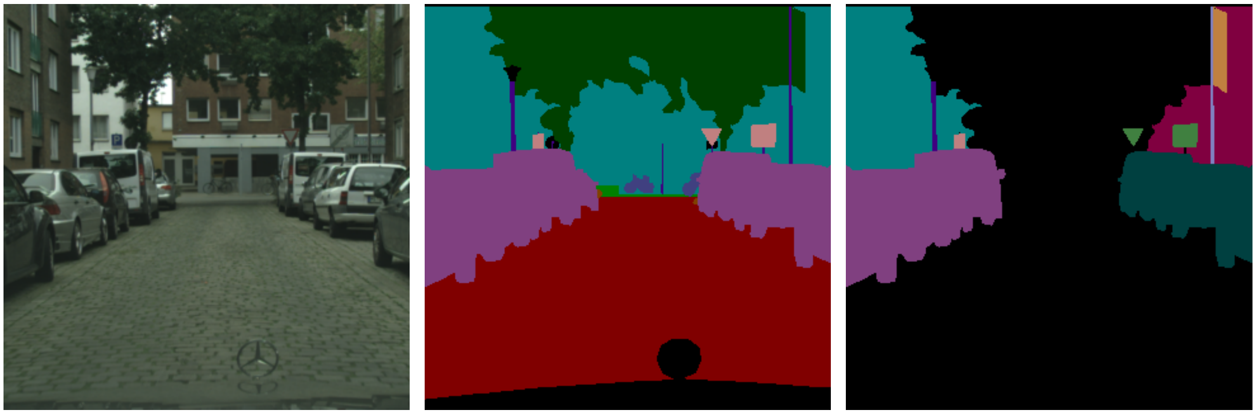


Figure 4.6: **Extracted eigenvectors on Cityscapes for both graph choices.** We visualize selected components of $\boldsymbol{X}_{\text{ortho}}$, sorted by decreasing eigenvalue. Three eigenvectors at a time are rendered as RGB images. In the Q-K case, eigenvectors detect the scene spatial layout and indicate how far left or right buildings, cars, trees, and people are. In the V-V case, eigenvectors perform semantic recognition and separate trees and buildings from road, and distinguish cars, people, and road markings.



(a) Image (b) Semantic Label (c) Processed Label

Figure 4.7: **Spatial semantic segmentation task.** We generate labels for "spatial semantic segmentation" by splitting the semantic labels into left/right subgroups, followed by filtering out small regions and ambiguous regions close to the image center.

Part II

REPRESENTATION LEARNING FOR GENERATIVE MODELS

CHAPTER 5
NESTED DIFFUSION MODELS USING HIERARCHICAL LATENT PRIORS

We introduce nested diffusion models, an efficient and powerful hierarchical generative framework that substantially enhances the generation quality of diffusion models, particularly for images of complex scenes. Our approach employs a series of diffusion models to progressively generate latent variables at different semantic levels. Each model in this series is conditioned on the output of the preceding higher-level models, culminating in image generation. Hierarchical latent variables guide the generation process along predefined semantic pathways, allowing our approach to capture intricate structural details. To construct these latent variables, we leverage a pre-trained visual encoder, which learns strong semantic visual representations, and modulate its capacity via dimensionality reduction and noise injection. Across multiple datasets, our system demonstrates significant enhancements in image quality for both unconditional and class/text conditional generation. Moreover, our unconditional generation system substantially outperforms the baseline conditional system. These advancements incur minimal computational overhead as the more abstract levels of our hierarchy work with lower-dimensional representations.

5.1 Introduction

The modern era of computer vision opened with deep networks driving advances in representation learning: mapping images, patches, or pixels to feature vectors that encode semantic information and support a range of downstream tasks such as classification [146, 222, 95], segmentation [168, 93], and object detection [82, 96, 32]. A variety of deep generative methods have since emerged to enable the reverse mapping, from a given prior or a learned embedding, back to the space of images. GANs [85], VAEs [136, 225, 245, 191, 170], normalizing flows [187, 1, 258], and diffusion models [89, 88, 284, 226] have demonstrated capacity to synthesize complex real-world image, video, and language data [9, 185, 166]. In parallel, representation learning has advanced through development of scalable architectures and training objectives, yielding self-supervised approaches, including contrastive learning [45, 40, 97, 34], masked autoencoders (MAEs) [98], and hybrids [296], that rival supervised feature learning.

Although generation and representation learning may have different immediate applications, they are inherently linked. A process that synthesizes realistic images must internally capture some notion of semantics in order to produce globally coherent structure. Indeed, recent work investigating generative models reveals that they capture rich visual representations useful for downstream tasks: segmentation [11, 269], image intrinsics [67] and image recognition [272, 152]. Conversely, another branch of research demonstrates that strong visual representations can further enhance generation quality via: conditioning on clustered features [108], learning to generate visual features that serve as a conditional signal [157], or adding a self-supervised representation learning loss to a generative model [156, 276]. However, these uses of pre-trained visual encoders or feature learning objectives focus only on abstract, high-level features, and in essence may function as unsupervised substitutes for image category labels.

Images contain diverse, multi-scale structures, from local textures and edges to parts, objects, and coherent scenes. For a generative system to produce realistic images, it must model all of these aspects. Current generative models frequently struggle to accurately represent attributes such as physical properties [121] and geometric layout [212], suggesting that conventional generative training objectives are insufficient for capturing these complex visual relationships.

To address this, we anchor a generation process to a visual feature hierarchy, which provides intermediate targets to guide progressive image synthesis. Our system employs a series of diffusion models, each operating at a different level of semantic abstraction and conditioned on outputs from higher levels. We build training targets for this hierarchical generator using a pre-trained visual encoder, applied to image patches of varying scale, in order to represent visual structures ranging from local texture to global shape. As additional controls on our target feature hierarchy, we

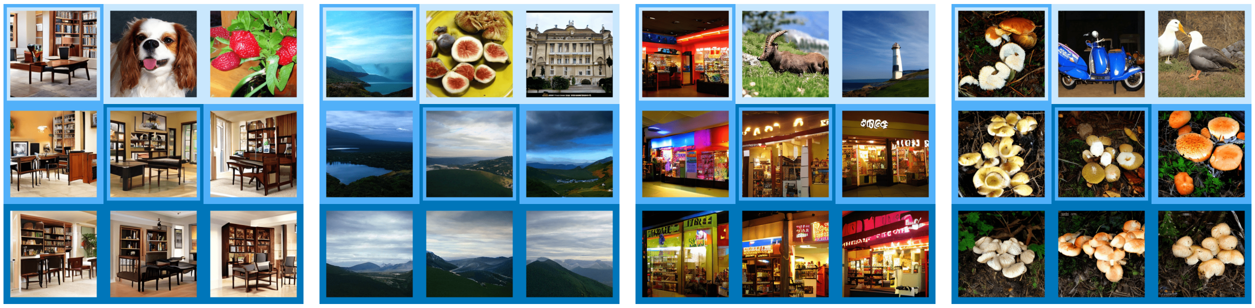


Figure 5.1: **Image generation via diffusion models nested along a hierarchical semantic chain.** We synthesize images using a sequence of diffusion models to generate a hierarchy of latent representations, starting from a low-dimensional semantic feature embedding and refining to a detailed image. At each hierarchical level, synthesis of a higher-dimensional latent from noise is conditioned on the more abstract latents generated at levels above. Here, each successive row visualizes resulting images when fixing latents up to some level and resampling those at subsequent levels; images (darker background) are produced by resampling only the more detailed levels of the hierarchical representation of a specific image in the preceding row (lighter background). Trained on ImageNet-1K, our multi-level generation system, *free of any external conditioning* (*i.e.*,, no class labels), learns a hierarchy that transitions from reflecting abstract semantic similarities to fine visual details.

compress feature representations through dimensionality reduction and noise-based perturbation. These capacity controls are essential to prevent memorization of image details at intermediate levels, allowing us to scale our model to deep hierarchies.

Unlike traditional methods using VAE features [204] or image pyramids [88] that primarily focuses on local textures, our approach emphasizes structured, multi-scale semantic representations. Compared to the hierarchical VAE [245, 293, 48, 236] models that run generation in a hierarchical latent space but suffer training instability, we use frozen latent representations, which significantly enhances training stability and yields much better generation quality.

Figure 5.1 shows example output using our method for unconditional synthesis on ImageNet-1k. Our unconditional system even outperforms the conditional generation baseline, as benchmarked in Figure 5.3, and also achieves consistent quality improvement as the number of levels L in the hierarchy increases from 2 to 5. Figure 5.2 sketches the key components of our model architecture. Section 5.4 extends experiments to the challenging setting of text-conditioned image generation trained on COCO scenes, where our hierarchical model outperforms baseline models containing substantially more parameters and consuming significantly larger training datasets. Our contributions are as follows:

- We introduce nested diffusion models, anchoring image generation to a hierarchical feature representation. Top hierarchy levels promote consistency in global image structure, while subsequent levels refine visual details. Resampling specific levels gives tunable control over synthesis.

- Our design greatly enhances generation quality while maintaining efficiency. Our five-level hierarchical model increases computational cost, measured in GFlops, by only 25% relative to single-level diffusion, yet significantly improves quality. Relative to a baseline model requiring comparable GFlops, we decrease FID from 45.19 to 11.05 for unconditional generation and from 31.13 to 9.87 for conditional generation on ImageNet-1k.

- Our system consistently improves performance in both conditional and unconditional generation tasks as more hierarchical levels are added. Notably, on ImageNet-1k, our unconditional generation quality surpasses that of the baseline class-conditional diffusion model.

5.2 Related Work

Hierarchical models. Hierarchical variational autoencoders (HVAEs) [245, 293, 48, 236] extend the latent space of VAEs [136] to include multiple variables, and demonstrate improved generation quality. However, HVAEs are known to suffer from high variance and collapsed representations, where the top-level variables may be ignored [245, 48]. To address this issue, Luhman and Luhman [170] introduce a layer-wise scheduler and regularization to enhance stability, while Hazami et al. [94] propose a simplified architecture.

Recent work has sought to build hierarchical generative systems by freezing the latent variables and leveraging powerful generative methods such as diffusion models and autoregressive models. For example, Ho et al. [106], Gu et al. [88], Liu et al. [166] train a set of diffusion models to handle

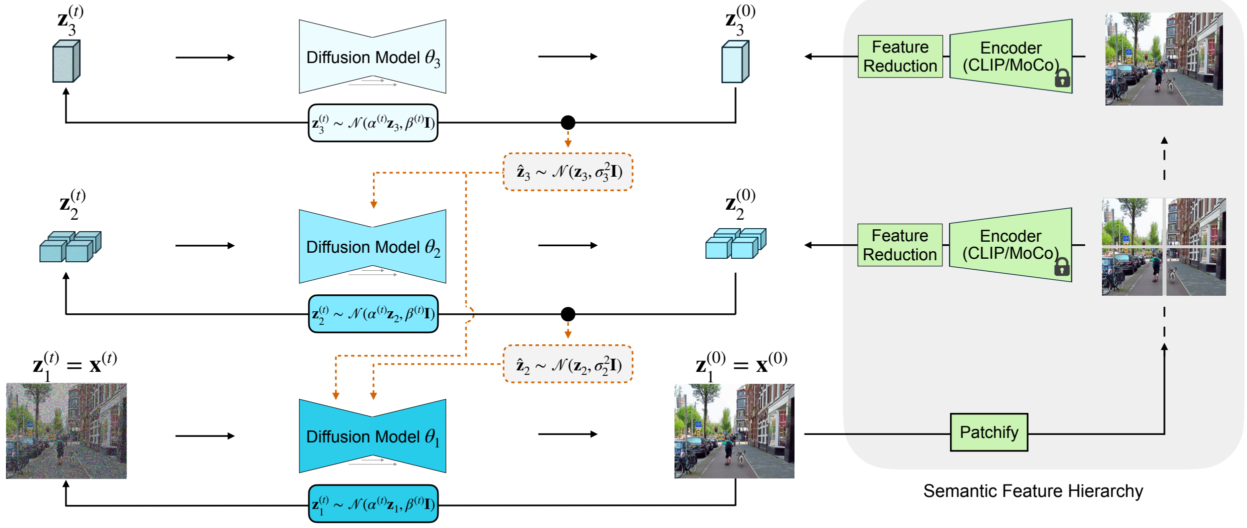

$\mathbf{z}_3^{(t)}$
Diffusion Model θ_3
$\mathbf{z}_3^{(0)}$
$\mathbf{z}_3^{(t)} \sim \mathcal{N}(\alpha^{(t)}\mathbf{z}_3, \beta^{(t)}\mathbf{I})$
$\hat{\mathbf{z}}_3 \sim \mathcal{N}(\mathbf{z}_3, \sigma_3^2\mathbf{I})$
$\mathbf{z}_2^{(t)}$
Diffusion Model θ_2
$\mathbf{z}_2^{(0)}$
$\mathbf{z}_2^{(t)} \sim \mathcal{N}(\alpha^{(t)}\mathbf{z}_2, \beta^{(t)}\mathbf{I})$
$\hat{\mathbf{z}}_2 \sim \mathcal{N}(\mathbf{z}_2, \sigma_2^2\mathbf{I})$
$\mathbf{z}_1^{(t)} = \mathbf{x}^{(t)}$
Diffusion Model θ_1
$\mathbf{z}_1^{(0)} = \mathbf{x}^{(0)}$
$\mathbf{z}_1^{(t)} \sim \mathcal{N}(\alpha^{(t)}\mathbf{z}_1, \beta^{(t)}\mathbf{I})$
Feature Reduction
Encoder (CLIP/MoCo)
Feature Reduction
Encoder (CLIP/MoCo)
Patchify
Semantic Feature Hierarchy

Figure 5.2: **Nested diffusion architecture.** ***Left:*** We train a sequence of diffusion models to generate a hierarchical collection of latent representations $\{\boldsymbol{z}_3, \boldsymbol{z}_2, \boldsymbol{z}_1 = \boldsymbol{x}\}$ of increasing dimensionality up to an image $\boldsymbol{z}_1 = \boldsymbol{x}$. Generated latents serve as conditional inputs (dotted lines) to diffusion models at subsequent levels, with separately parameterized noising processes, $\hat{\boldsymbol{z}}_l \sim \mathcal{N}(\boldsymbol{z}_l, \sigma_l^2\mathbf{I})$, controlling the information capacity of these signals. ***Right:*** A pre-trained, frozen visual encoder provides target latent representations for each level of the hierarchy. To construct these latent features, we run the encoder on patchified images, reducing patch size and applying dimensionality reduction across feature channels in order to shift focus from local details to global semantics. Upper level targets encode more abstract semantics and, being lower-dimensional vectors, are less computationally expensive to synthesize, making hierarchical generation fast.

images at different resolutions, and Tian et al. [242] train a hierarchical autoregressive model to predict the residuals between tokenized representations at adjacent resolutions. However, none of these approaches involve training with hierarchical semantic latent representations.

Conditional generation. A conditional diffusion model aims to parameterize the prior as a complex joint distribution conditioned on an input, rather than using a simple Gaussian prior, significantly enhancing the model's capacity to capture intricate data patterns. For images of complex scenes, generation conditioned on image captions [89, 122, 201] has shown notable improvements in both quality and controllability. [284, 204] extend this conditioning approach to multiple modalities, incorporating input such as segmentation, depth maps, and human joint positions. Another direction in this field is learning the conditional variable itself. Models like DiffAE [193], SODA [112], and Abstreiter et al. [2] train an encoder to produce a low-dimensional latent vari-

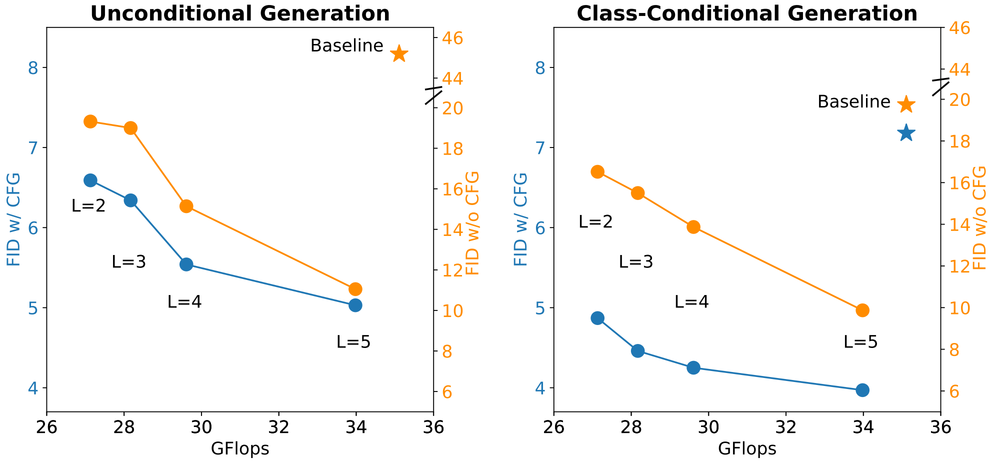


Figure 5.3: **Image generation quality when scaling our nested diffusion models on ImageNet-1K dataset.** The deeper hierarchies we build lead to a slight increase in computational overhead (particularly when $L \leq 4$), as measured by GFlops, while significantly improving the generation quality. Compared to the single-level baseline model using comparable GFlops, our 5-level unconditional system significantly improves the performance w/o classifier-free guidance (CFG) by reducing FID from 45.19 to 11.05, exceeding the class-conditional baseline of 19.74.

able to assist the generation process; these works also demonstrate that such an encoder can learn meaningful image representations.

Generation with semantic visual representations. State-of-the-art generative models, such as diffusion and autoregressive models, can be viewed as denoising autoencoders that inherently learn meaningful data representations. Yang and Wang [272], Tang et al. [238], Zhang et al. [291] demonstrate that diffusion models capture semantic visual representations, which are directly applicable to various downstream tasks [11, 124]. Zhang and Maire [288] highlight that a discriminator in a GAN can learn useful representations. [156, 119] show that incorporating representation learning objectives into the generative framework can further enhance generation quality. Li et al. [157], Hu et al. [108], Wang et al. [256] leverage semantic representations learned by the encoder to further improve generation quality.

5.3 Method

We employ a structured approach to capture hierarchical semantic features for image generation.

5.3.1 Preliminary: Diffusion models

As a generative framework, diffusion models [105, 228, 226] consist of both a forward (diffusion) process and a backward process, each spanning over T steps. Let $\mathbf{x} \in \mathbb{R}^d$ denote the original data sample. The forward process defines a sequence of latent variables $\{\boldsymbol{x}^{(t)}\}_{t=1}^{T}$ obtained by sampling from a Markov process parameterized as $q\left(\boldsymbol{x}^{(t)} \mid \boldsymbol{x}^{(t-1)}\right) := \mathcal{N}(\boldsymbol{x}^{(t)}; \alpha^{(t)}\boldsymbol{x}, \beta^{(t)}\mathbf{I})$, where $\alpha^{(t)}$ and $\beta^{(t)}$ are hyperparameters of the noise scheduler, ensuring that the signal-to-noise ratio (SNR) decreases as t increases.

In the backward process, the model $\boldsymbol{D}_\theta$ is tasked with estimating the transition probability $p(\boldsymbol{x}^{(t-1)}|\boldsymbol{x}^{(t)})$ and generating data through the process $\prod_{t=1}^{T} p_\theta(\boldsymbol{x}^{(t-1)}|\boldsymbol{x}^{(t)})p(\boldsymbol{x}^{(T)})$, where $p_\theta(\boldsymbol{x}^{(t-1)}|\boldsymbol{x}^{(t)})$ represents the transition probability estimated by $\boldsymbol{D}_\theta$. It is trained by optimizing the Variational Lower Bound (VLB) [134]:

$$\mathcal{L}_{\text{VLB}} = \sum_{t=1}^{T} D_{\text{KL}}\left(q\left(\boldsymbol{x}^{(t-1)}|\boldsymbol{x}^{(t)}, \boldsymbol{x}\right) \middle\| p_\theta\left(\boldsymbol{x}^{(t-1)}|\boldsymbol{x}^{(t)}\right)\right). \tag{5.1}$$

Here $q\left(\boldsymbol{x}^{(t-1)}|\boldsymbol{x}^{(t)}, \boldsymbol{x}\right)$ can be derived using Bayes' rule as $q\left(\boldsymbol{x}^{(t)}|\boldsymbol{x}^{(t-1)}, \boldsymbol{x}\right) q\left(\boldsymbol{x}^{(t-1)}|\boldsymbol{x}\right) / q\left(\boldsymbol{x}^{(t)}|\boldsymbol{x}\right)$. Minimizing the RHS of Eqn.5.1 can be simplified as training $\boldsymbol{D}_\theta$ to estimate the noise $\boldsymbol{\epsilon}^{(t)} \in \mathbb{R}^d$ sampled from $\mathcal{N}(0, \mathbf{I})$ [105]:

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{\boldsymbol{\epsilon}^{(t)}, t} \|\boldsymbol{D}_\theta(\alpha^{(t)}\boldsymbol{x} + \beta^{(t)}\boldsymbol{\epsilon}^{(t)}, t) - \boldsymbol{\epsilon}^{(t)}\|_2.$$

5.3.2 Nested diffusion models

We propose a hierarchical generative framework with L levels, where each level is instantiated as a diffusion model $\boldsymbol{D}_{\theta_l}$. As illustrated in Figure 5.2, the generative model at level l produces latent variable $\boldsymbol{z}_l \in \mathbb{R}$ with $p_{\theta_l}(\boldsymbol{z}_l|\boldsymbol{z}_{>l})$, where $\boldsymbol{z}_{>l} := \{\boldsymbol{z}_m\}_{m>l}$ represents latent variables from higher levels. The feature dimension of $\boldsymbol{z}_l \in \mathbb{R}^{d_l}$ decreases as l increases, such that $d_l > d_{l+1}$. At the shallowest level when $l = 1$, the latent variables correspond directly to the data samples; that is, $\mathbf{z}_1 = \mathbf{x}$.

Diffusion with semantic hierarchy. Our approach explicitly guides the generative process to align with a semantic hierarchy. Here, the top tier (larger l) denotes higher levels of semantic abstraction, whereas the lower tier (smaller l) represents detailed, fine-grained information. This is essential for preserving image semantic structures and producing realistic samples in generative models.

Non-Markovian generation. At each hierarchical level l, we follow the diffusion model framework and task $\boldsymbol{D}_{\theta_l}$ to estimate the transition probability $p_{\theta_l}(\boldsymbol{z}_l^{(t-1)}|\boldsymbol{z}_l^{(t)}, \boldsymbol{z}_{>l})$. At layer l, we assume a non-Markovian generation process where $\boldsymbol{D}_{\theta_l}$ depends on the entire set of latent variables $\boldsymbol{z}_{>l}$ estimated from the preceding hierarchies.

We optimize the hierarchal diffusion models using the following objectives:

$$\begin{aligned}&\mathcal{L}_{\text{hierarchical\_ELBO}} = \\ &\sum_{l=1}^{L-1}\sum_{t=1}^{T} D_{\text{KL}}\left(q\left(\boldsymbol{z}_l^{(t-1)}|\boldsymbol{z}_l^{(t)}, \boldsymbol{z}_l, \boldsymbol{x}\right)\middle\|p_{\theta_l}\left(\boldsymbol{z}_l^{(t-1)}|\boldsymbol{z}_l^{(t)}, \boldsymbol{z}_{>l}\right)\right) \\ &+\sum_{t=1}^{T} D_{\text{KL}}\left(q\left(\boldsymbol{z}_L^{(t-1)}|\boldsymbol{z}_L^{(t)}, \boldsymbol{z}_L, \boldsymbol{x}\right)\middle\|p_{\theta_l}\left(\boldsymbol{z}_L^{(t-1)}|\boldsymbol{z}_L^{(t)}\right)\right).\end{aligned} \tag{5.2}$$

Comparing to hierarchical VAEs which also includes hierarchical latent variables $\{\boldsymbol{z}_l\}_{l=1}^{L}$, we enhance sampling capability by integrating the diffusion model and introducing an additional set of latent variables $\{\boldsymbol{z}_l^{(t)}\}_{t=0}^{T}$ for each level l. This modification allows for multiple sampling

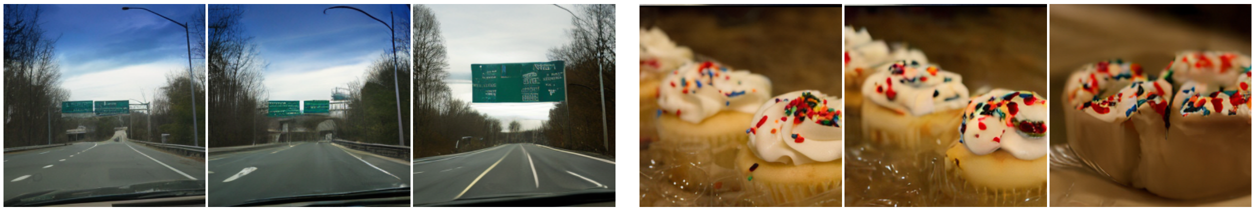


Image $\sigma_2 = 0$ $\sigma_2 = 0.5$ Image $\sigma_2 = 0$ $\sigma_2 = 0.5$

Figure 5.4: **Feature compression via Gaussian noise.** For a two-level hierarchical generator ($L = 2$), we generate images conditioned on an oracle CLIP feature $\boldsymbol{z}_2$, inferred from input images, with feature channels reduced from 512 to 256 dimensions via SVD. Without noise ($\sigma_2 = 0$) added to $\boldsymbol{z}_2$, the generator $\boldsymbol{D}_{\theta_1}$ degenerates to an autoencoder that nearly reconstructs the input; adding Gaussian noise ($\sigma_2 = 0.5$) to $\boldsymbol{z}_2$ limits feature information, allowing for generation of new content.

steps, as opposed to the single forward pass used in hierarchical VAEs, leading to a more accurate prior estimation. This improvement is vital in hierarchical generative systems, where mismatches between the posterior and prior distributions can compound across levels, potentially degrading the quality of the generated output.

5.3.3 Hierarchical latents & progressive compression

Extraction of hierarchical features. We construct $\{\boldsymbol{z}_l\}_{l=1}^{L}$ from a pre-trained visual encoder and keep those frozen during training. Most visual encoders, denoted as $\boldsymbol{E}(\boldsymbol{x}) = \boldsymbol{z}$, map an input image $\boldsymbol{x}$ to a vector $\boldsymbol{z}$. To build a hierarchical representation, we propose running $\boldsymbol{E}$ on image patches. Let $C(\boldsymbol{x}, M) = \{\boldsymbol{x}_m\}_{m=1}^{M^2}$ represent the cropping operation that splits the image $\boldsymbol{x} \in \mathbb{R}^{H \times W \times 3}$ into M^2 non-overlapping square patches $\boldsymbol{x}_m \in \mathbb{R}^{\frac{H}{M} \times \frac{W}{M} \times 3}$. We build a latent variable $\boldsymbol{z}_l = \boldsymbol{E}\left(C\left(\boldsymbol{x}, L - l + 1\right)\right)$ over image patches. As patch size decreases, the visual representation transitions from capturing global structures to more localized features.

Feature compression via Gaussian noise. Our construction of $\{\boldsymbol{z}_l\}_{l=1}^{L}$ contrasts with hierarchical VAEs, which use compression objectives to learn hierarchical latent variables but often face high variance issues, as noted in prior work [191, 245, 48]. While we address instability, our design presents a new challenge: representations from the visual encoder tend to be highly informative, allowing the generative model to reconstruct the input image accurately, which can cause the generator to behave like an autoencoder.

We show this effect in Figure 5.4: when conditioning on the oracle CLIP visual features, the diffusion model could nearly reconstruct the input with a delta distribution: $p_{\theta_L}(\boldsymbol{z}_0|\boldsymbol{z}_L) \approx \delta(\boldsymbol{z}_0)$, essentially “bypassing” the middle levels $p_{\theta_l}(\boldsymbol{z}_l|\boldsymbol{z}_{l+1})$ in a hierarchical system. Consequently, we must reduce the information contained in $\boldsymbol{z}_l$ to ensure each hierarchical level contributes meaningfully to the generation process; our procedure is as follows.

Channel reduction via singular value decomposition. In our patch-based approach, the channel number of the latent variable quadratically increases as we move down the hierarchy, with $\boldsymbol{z}_l \in \mathbb{R}^{(L-l+1)^2 \times d}$, quickly making the information overcomplete for generation. To address this, we propose trimming the feature channels. Specifically, we apply singular value decomposition (SVD) to the encoder's feature vector, preserving only the leading $d/(L-l+1)$ channels, resulting in $\boldsymbol{z}_l \in \mathbb{R}^{(L-l+1) \times d}$, with channel linearly increasing over level of hierarchy.

Information reduction through Gaussian noise. As shown in Figure 5.4, channel reduction alone is insufficient to prevent the diffusion model from degrading into an autoencoder. To further increase the abstraction level, we introduce Gaussian noise to $\boldsymbol{z}_l$, represented as $\hat{\boldsymbol{z}}_l \sim \mathcal{N}(\boldsymbol{z}_l, \sigma_l^2 \mathbf{I})$, where σ_l is a fixed constant based on the hierarchical level. This process limits the amount of information that can be transmitted, measured by the KL divergence $D_{KL}\left(\mathcal{N}\left(\boldsymbol{z}_l, \sigma_l^2\right), \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)\right)$. A large variance σ_l^2 substantially limits the information capacity. In our experiments, adding Gaussian noise proved essential in preserving and enhancing generation quality as the number of hierarchical levels increased. We only add Gaussian noise during training; it is not present during generation.

With these approaches, our loss function is:

$$
\begin{aligned}
&\mathcal{L}_{\text{nested\_diffusion}} = \\
&\sum_{l=1}^{L-1} \mathbb{E}_{\hat{\boldsymbol{z}}_{>l}, \boldsymbol{\epsilon}_l^{(t)}, t} \| \boldsymbol{D}_{\theta_l}(\alpha^{(t)} \boldsymbol{z}_l + \beta^{(t)} \boldsymbol{\epsilon}_l^{(t)}, \hat{\boldsymbol{z}}_{>l}, t) - \boldsymbol{\epsilon}_l^{(t)} \|_2 \\
&+ \mathbb{E}_{\boldsymbol{\epsilon}_L^{(t)}, t} \| \boldsymbol{D}_{\theta_L}(\alpha^{(t)} \boldsymbol{z}_L + \beta^{(t)} \boldsymbol{\epsilon}_L^{(t)}, t) - \boldsymbol{\epsilon}_L^{(t)} \|_2,
\end{aligned}
\tag{5.3}
$$

where $\boldsymbol{\epsilon}_l^{(t)} \in \mathbb{R}^{d_l}$ denotes noise sampled at each level.

5.3.4 Diffusion with semantic consistent neighbors

Diffusion models are highly related to the *mean-shift* iterations [52, 47], as highlighted by the analysis [253, 227, 131]. Specifically, assuming hierarchical dependency $p(\boldsymbol{z}_l^{(t)} | \boldsymbol{z}_{>l}) = \mathbb{E}_{\boldsymbol{z}_l} p(\boldsymbol{z}_l^{(t)} | \boldsymbol{z}_l, \boldsymbol{z}_{>l}) = \mathbb{E}_{p(\boldsymbol{z}_l | \boldsymbol{z}_{>l})} p(\boldsymbol{z}_l^{(t)} | \boldsymbol{z}_l)$, an optimal denoiser $\boldsymbol{D}_l^*$ for Eqn. 5.3 can be expressed in closed-form as follows [227, 131]:

$$
\boldsymbol{D}_l^*(\boldsymbol{z}_l^{(t)} | \boldsymbol{z}_{>l}) = \frac{\int_{\boldsymbol{z}_l} p(\boldsymbol{z}_l^{(t)} | \boldsymbol{z}_l) p(\boldsymbol{z}_l | \boldsymbol{z}_{>l}) \boldsymbol{z}_l}{\int_{\boldsymbol{z}_l} p(\boldsymbol{z}_l^{(t)} | \boldsymbol{z}_l) p(\boldsymbol{z}_l | \boldsymbol{z}_{>l})}. \tag{5.4}
$$

This suggests that the optimal solution is the weighted data point $\boldsymbol{z}_l$, based on the similarity between $\boldsymbol{z}_l$ and $\boldsymbol{z}_l^{(t)}$. Therefore, the structure of $\boldsymbol{z}_l$ significantly impacts the quality of optimal denoiser. Ideally, the neighbor of both $\boldsymbol{z}_l^{(t)}$ and $\boldsymbol{z}_l$ should have similar semantic structures. In Figure 5.5, we attempt to visualize the structure of $\boldsymbol{z}_l$ via nearest neighbor images with CLIP features or VAE features. Unlike VAE, which focuses on low-level textures and results in unrelated neighboring images, CLIP yields semantically similar images and better generation quality in our experiments.

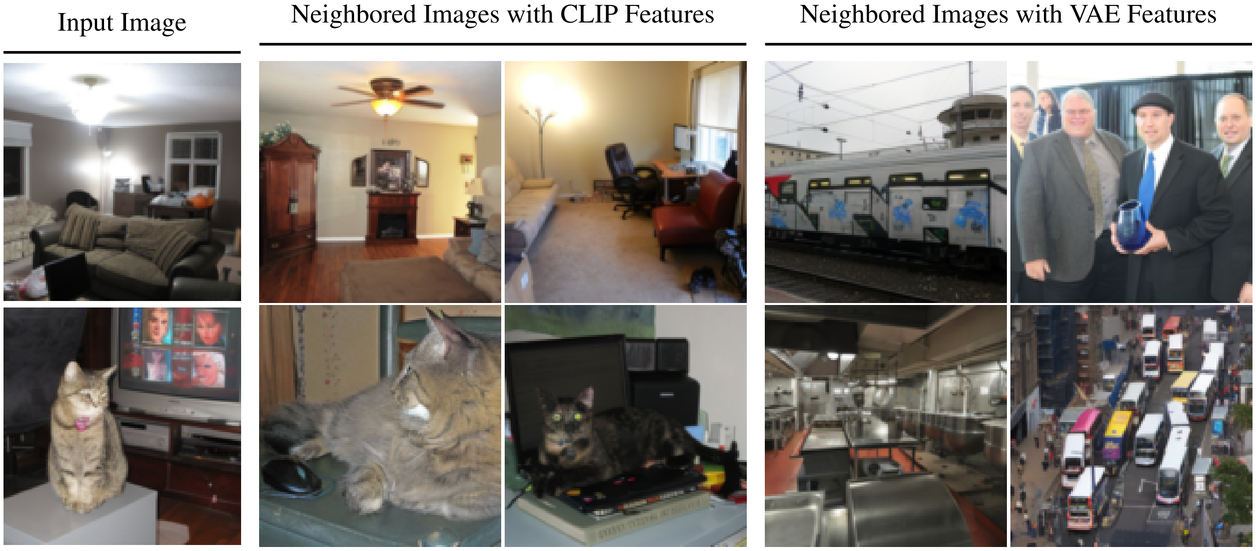

Input Image
Neighbored Images with CLIP Features
Neighbored Images with VAE Features

Figure 5.5: **Visualization of K-Nearest Neighbors (KNN) with different sources of latent features.** For each input image, we display neighboring images, based on features extracted from two types of visual representations: CLIP representations, and VAE bottlenecks. Unlike the VAE, which focuses on low-level visual structures, CLIP emphasizes semantic representations, yielding more meaningful nearest neighbors. Our experiments demonstrate that running a diffusion model on a latent space with well-structured neighbors is essential for enhancing generation quality.

5.4 Experiments

We present the setup and results of our experiments, where we evaluate the performance of our nested diffusion model across various tasks. Our primary focus is to explore the model's effectiveness in both conditional and unconditional image generation scenarios using the COCO-2014 [163] and ImageNet-1K datasets [208].

5.4.1 Experimental Setup

Nested diffusion models. We utilize U-ViT [9], a ViT-based U-Net model with an encoder-decoder architecture, for $\boldsymbol{D}_{\theta_l}$. This model employs skip connections and performs diffusion in the latent space of a pre-trained VAE, reducing the input size from $256 \times 256 \times 3$ to $32 \times 32 \times 4$, which enables efficient handling of high-resolution images. We customize the network configurations to make it aligned with the standard ViT-Base model: The transformer model we use consists

	w/o CFG			w/ CFG		
Model	σ_2 = 0.0	0.5	1.0	σ_2 = 0.0	0.5	1.0
$L = 1$	55.41	-	-	-	-	-
$L = 1^*$	45.19	-	-	-	-	-
$L = 2$	**19.32**	20.66	27.40	**6.59**	7.19	8.69
$L = 3$	20.34	**19.00**	23.37	6.77	**6.34**	6.98
$L = 4$	17.67	**15.14**	16.27	5.79	**5.54**	5.89
$L = 5$	19.04	11.88	**11.05**	7.68	5.36	**5.03**

(a) *Unconditional* image generation for ImageNet-1k, 256×256

	w/o CFG			w/ CFG		
Model	σ_2 = 0.0	0.5	1.0	σ_2 = 0.0	0.5	1.0
$L = 1$	31.13	-	-	13.75	-	-
$L = 1^*$	19.74	-	-	7.18	-	-
$L = 2$	**16.56**	**16.52**	22.43	**4.87**	5.31	6.49
$L = 3$	**15.51**	**15.50**	16.35	**4.46**	4.69	5.15
$L = 4$	17.72	14.38	**13.87**	4.81	4.38	**4.25**
$L = 5$	18.04	11.28	**9.87**	4.26	4.05	**3.97**

(b) *Conditional* image generation for ImageNet-1k, 256×256

Table 5.1: We evaluate image generation quality using the Fréchet Inception Distance (FID) on the ImageNet-1k and we report the results w/o and w/ classifier free guidance (CFG). We benchmark our model across different noise levels and network depths L. To determine the noise levels $\{\sigma_l\}_{l=2}^{L}$, we use a top-down, greedy searching: for a model with depth L, we retain the optimal values $\{\sigma_l\}_{l=3}^{L}$ from the shallower model $< L$ and only tune the newly added level, σ_2. The generation quality improves with L increases and adding Gaussian noise is crucial for better performance for deeper model. For comparison, we also provide baseline results for $L = 1^*$, a single-level model with increased parameters to match the GFLOPs of $L = 5$.

of 12 blocks, with the base channel dimension set to 768, and each attention block comprises 12 attention heads. We use the default diffusion scheduler, sampler, and training hyperparameters from U-ViT [9] for ImageNet-1k and COCO respectively.

To construct our nested diffusion model, we use the same network configuration for each hierarchical level. At higher hierarchical levels, there is a progressive reduction in the dimensionality of $\boldsymbol{z}_l$, which leads to minimal extra computational cost even though the number of parameters increases.

To incorporate conditional features from higher levels, we simply treat $\hat{\boldsymbol{z}}_{l+1}$ as an additional input token and append it to $\boldsymbol{z}_l$ before feeding into the ViT. During training, we randomly (10%

chance) replace the $\hat{\boldsymbol{z}}_{l+1}$ with a learnable empty token to facilitate classifier-free guidance (CFG) [104]. We use the same architecture for both COCO and ImageNet-1k experiments and we train on COCO for 1000 epochs and ImageNet-1k for 200 epochs unless mentioned otherwise.

We follow the standard evaluation protocol to report the image generation quality with Fréchet inception distance (FID). For ImageNet-1k, we generate 50K images, 50 for each category, and compute the FID over the training set using the precomputed statistic provided by Dhariwal and Nichol [59]. For COCO-2014, when comparing to other approaches in Table 5.4, we follow previous literature to generate 30k images using text prompts from the validation set and compare the statistics against the validation set images.

Hierarchical latent variables. We build hierarchical latent variables $\{\boldsymbol{z}_l\}_{l=2}^{L}$ using a pre-trained visual encoder and directly use VAE bottleneck features as our bottom level $\boldsymbol{z}_1$. For the ImageNet experiments, we extract visual features from MoCo-v3 [45] (ViT-B/16), a leading self-supervised visual representation learner. For the COCO experiments, we use CLIP [195] (ViT-B/16), a multi-modal encoder that aligns visual and textual representations.

We set the top-level latent representation to a fixed dimension: $\boldsymbol{z}_L \in \mathbb{R}^{256}$, by trimming additional feature channels via singular value decomposition (SVD). Consequently, we construct hierarchical latent variables with shapes: $\boldsymbol{z}_L \in \mathbb{R}^{1\times 256}$, $\boldsymbol{z}_{L-1} \in \mathbb{R}^{4\times 128}$, $\boldsymbol{z}_{L-2} \in \mathbb{R}^{16\times 64}$, and so on.

Patchification enables us to extract features at various resolutions from the encoder. However, this approach can encounter limitations when patches become too small to carry meaningful visual patterns. For cases requiring patches' spatial resolution smaller than 64×64, we use the feature maps from the encoder's backbone instead. For example, with $L = 5$, $\boldsymbol{z}_2 \in \mathbb{R}^{64\times 32}$ is built by reducing the feature map of the encoder, with shape $14 \times 14 \times 768$ (height $\times$ width $\times$ channel numbers) to $8 \times 8 \times 32 = 64 \times 32$ through spatial pooling and channel reduction.

Efficient training and parameter search for hierarchical models. In our design, we allow each $\boldsymbol{D}_{\theta_l}$ to have a unique σ_l, providing flexibility, but also adding complexity to hyperparameter

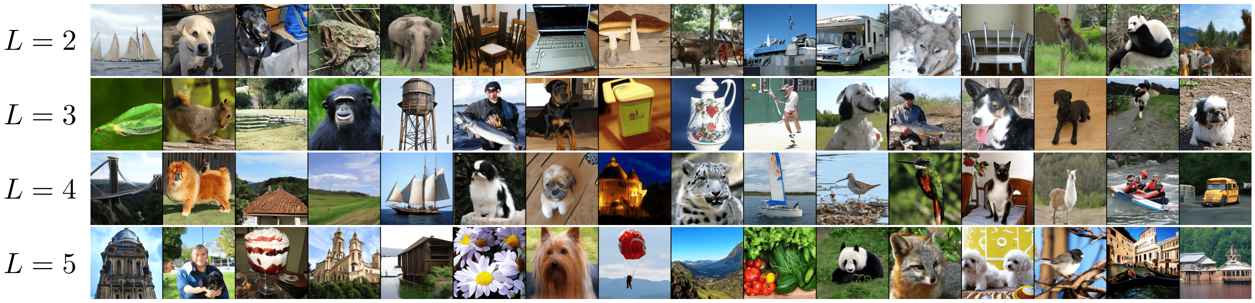


Figure 5.6: **Visualization of unconditional image generation on ImageNet-1K.** We present visualizations of images generated by hierarchical diffusion models containing from 2 to 5 levels, demonstrating that image quality improves as the depth of the hierarchy increases.

Model	Iter.	GFlops	w/ CFG	w/o CFG
Conditional Generation				
DiT-XL/2 [190]	7M	118.6	2.27	9.62
DiT-XL/2 [190]	400k	118.6	n/a	19.95
DiT-XL/2 + REPA [276]	400k	118.6	n/a	12.30
DiT-XL/2 + REPA [276]	850k	118.6	n/a	9.60
U-ViT (Mid)	500k	26.8	13.75	31.13
U-ViT (Mid*)	500k	35.1	7.18	19.74
Ours ($L = 5$)	500k	34.0	**3.97**	**9.87**
Unconditional Generation				
U-ViT (Mid)	500k	26.8	n/a	55.41
U-ViT (Mid*)	500k	35.1	n/a	45.19
Ours ($L = 5$)	500k	34.0	**5.03**	**11.05**

Table 5.2: **Generation quality (FID) on ImageNet-1k 256×256.** Our model significantly outperforms U-ViT (Mid) and U-ViT (Mid*), which are $L = 1$ and $L = 1*$ from Table 5.1. We show in grayscale results requiring substantially more training iterations or more computation as measured in GFlops.

search. To streamline this process, we use a hierarchical local search strategy: for a L-level model, we retain the optimal noise levels $\{\sigma_l\}_{l=3}^{L}$ from the $L - 1$ level model and search only for the newly introduced level σ_2. An additional benefit of this approach is that we can directly reuse the parameters $\{\boldsymbol{D}_{\theta_l}\}_{l=3}^{L}$ from shallower models due to the consistent model configuration, which means that we only need to train $\{\boldsymbol{D}_{\theta_l}\}_{l=1}^{2}$.

γ	0	0.1	0.3	0.5	∞
w/ CFG	5.09	5.05	**5.03**	5.33	11.21
w/o CFG	14.19	13.53	12.51	11.93	**11.27**

Table 5.3: We investigate the effects of different values of γ, which controls the level of noise added to $\boldsymbol{z}_l$ during generation through the term $(t/T)^\gamma \sigma_l$ at diffusion steps t, as described in Sec. 5.4.2. Here, we present the generation results (FID) using a five-level nested diffusion model ($L = 5$) on ImageNet-1K. When $\gamma = 0$, the model applies σ_l (consistent with the noise level during training) for generation, while $\gamma = \infty$ corresponds to no noise being added to $\boldsymbol{z}_l$ during generation.

5.4.2 Generation with noisy hierarchical features

During training, we introduce noise to hierarchical features $\boldsymbol{z}_l$ by sampling $\hat{\boldsymbol{z}}_l \sim \mathcal{N}(\boldsymbol{z}_l, \sigma l^2 \mathbf{I})$, as outlined in Section 5.3.3, to enhance information compression. By default, this noise is removed during generation to ensure consistent conditional signals. However, we observe that this approach can be detrimental to classifier-free guidance (CFG), likely due to a distributional shift between training and testing. To address this, we propose a gradual decay of σ_l across diffusion steps t during generation: $\hat{\boldsymbol{z}}_l^{(t)} \sim \mathcal{N}(\boldsymbol{z}_l, (t/T)^\gamma \cdot \sigma_l^2 \mathbf{I})$, where $0 \leq (t/T)^\gamma \leq 1$ progressively reduces σ_l, and the scalar $\gamma \geq 0$ controls the decay speed. This design shares the similar spirit of Sadat et al. [209], which adds noise to the ground truth image label to encourage diversity in the generation process. In contrast, our approach focuses on maintaining consistency between the training and testing phases.

Table 5.3 presents the results for various γ values in a $L = 5$ nested diffusion model for unconditional generation on ImageNet-1K. Setting $\gamma = 0$ applies the same noise level σ_l used during training, while $\gamma = \infty$ eliminates noise during generation. In experiments with CFG, adding noise to $\boldsymbol{z}_l$ is crucial, and our proposed scheme improves generation quality compared to the baseline of using the training noise level ($\gamma = 0$). Without CFG, the best results are achieved by omitting noise from $\boldsymbol{z}_l$. For subsequent experiments, we use $\gamma = 0.3$ in CFG scenarios and $\gamma = \infty$ in non-CFG cases.

5.4.3 Benchmarking generation quality

We present our primary results for ImageNet-1k in Table 5.1, including the choices of model depth L, the noise level, and generation w/ or w/o CFG.

Improved performance with more hierarchy levels L. Compared to the baseline model, our nested diffusion models demonstrate enhanced image quality as we deepen the hierarchical structure by increasing depth L. Specifically, our five-level generation significantly outperforms the baseline with $L = 1$ in both conditional generation ($31.13 \rightarrow 9.87$) and unconditional generation ($55.41 \rightarrow 11.05$). Notably, our unconditional generator with $L = 5$ surpasses the conditional baseline generation model with $L = 1$, achieving scores of $31.13 \rightarrow 11.05$.

We also present results using CFG, which directs the generated output toward conditional features, enhancing quality at the expense of reduced output diversity. The influence of CFG is controlled by the parameter w. Our nested diffusion models produce hierarchical representations; ideally, the top level benefits from stronger CFG to enforce semantic abstraction, while lower levels should reduce the CFG influence to promote diversity in visual details. To achieve this, we adopt a straightforward approach by specifying a set of decaying CFG weights, $\{w_i\} = [0.5, 0.4, 0.3, 0.2, 0.1]$, and selecting $\{w_i\}_{i=1:L}$ for our L-level nested diffusion models, with higher CFG weights assigned to the top levels. We conducted a limited hyperparameter search over the choices of CFG weights and found that this strategy yields better results than using a constant CFG weight for ImageNet conditional generation. For unconditional generation, we use a constant $w = 0.8$, as it produces better results.

With CFG, we demonstrate that increasing the hierarchy depth L further improves image quality, with our $L = 5$ model achieving FID scores of 3.97 for conditional generation and 5.03 for unconditional generation, significantly outperforming the baseline at $L = 1$ (13.75).

As illustrated in Figure 5.3, our hierarchical models achieve computational efficiency by constructing hierarchical features with decreased spatial dimensions, thereby reducing computational expense at higher tiers. In particular, our system with $L = 5$ results in only a 27.00% increase in the computational load measured in GFlops compared to the baseline $L = 1$, while achieving a marked decrease in FID by 68.29%, as detailed in Appendix D.

Impact of σ_l. As described in Section 5.3.3, σ_l plays a key role in enabling our design to scale effectively with more hierarchy levels. It regulates the information conveyed by the conditional latent variable z_l, ensuring that the hierarchical model does not bypass intermediate levels. We validate this design in Table 5.1, where the performance gap between $\sigma_2 = 0$ and non-zero σ_2 widens as the model depth L increases. This can be attributed to the fact that as L grows, more feature elements are included, increasing the likelihood of the model relying primarily on the lower-level features for generation, thus neglecting higher-level features. Higher noise levels help counteract this effect: with $L = 5$, setting $\sigma_2 = 1.0$ achieves FID of 11.05 and 9.87, outperforming $\sigma_2 = 0$, which yields FID of 18.04 and 19.04 for conditional and unconditional generation, respectively.

Comparison to other methods. We evaluate our models on class-conditional ImageNet 256×256 generation tasks without CFG, with results presented in Table 5.2. We compare against DiT [190] variants and REPA [276], which aligns diffusion representations with a pre-trained visual encoder and is trained for 400K steps. Our model with $L = 5$ significantly outperforms these baselines by a substantial margin while also requiring fewer GFlops. Remarkably, our unconditional model even surpasses their conditional version.

Experiments on COCO. In addition to ImageNet-1K, we also evaluate our model on the COCO-2014 dataset to assess performance on complex scenes. Using visual features from the CLIP ViT-B/16 model and fixed noise levels $\{\sigma_l\} = 0.5$, Table 5.4 reports results, comparing our approach with large models trained on additional data sources. Without CFG, our $L = 3$ system achieves state-of-the-art performance, surpassing most larger models. For our models, we set the CFG weight to 1, following the default in Bao et al. [9]. When generating with CFG, we find that $L = 2$ delivers the best performance, even outperforming the 7-billion-parameter model, CM3Leon. Our $L = 3$ model performs worse than $L = 2$, likely due to a suboptimal CFG weight.

Model	FID	Training Dataset
Huge Model, Extra Data		
GLIDE [185]	12.24	DALL-E (250M)
DALL-E 2 [200]	10.39	DALL-E (250M)
Imagen [210]	7.27	Internal Data/LAION (860M)
Re-Imagen [41]	5.25	KNN-ImageText/COCO (50M)
CM3Leon-7B [275]	4.88	Internal Data (350M)
Parti-20B [274]	**3.22**	LAION/FIT/JFT/COCO (4.8B)
COCO Data Only, w/ CFG		
VQ-Diffusion [89]	13.86	COCO (83K)
Friro [73]	8.97	COCO (83K)
U-ViT [9]	5.42	COCO (83K)
Ours $L = 2$	**4.72**	COCO (83K)
Ours $L = 3$	5.92	COCO (83K)
COCO Data Only, w/o CFG		
U-ViT [9]	14.98	COCO (83K)
Ours $L = 2$	8.15	COCO (83K)
Ours $L = 3$	**6.97**	COCO (83K)

Table 5.4: **Comparison of text-to-image generation on COCO-2014.** The upper half shows larger models trained with more data and the bottom half shows the models that are only trained on training split of COCO. When trained only on COCO, our models (with $\sigma_2 = 0.5$) outperform all the compared methods. It is worth noting that we're better than most of the larger models, shown on the top half.

5.4.4 Generation with different visual encoders

In Section 5.3.4, we discuss the importance of preserving neighbor structure for the target space of diffusion models. We quantitatively validate this claim in Table 5.5 by presenting results from a 3-level nested diffusion model ($L = 3$) with fixed $\{\sigma_l\} = 0.5$, applied to text-to-image generation on the COCO dataset. We construct $\boldsymbol{z}_l$ using various visual encoders, including MAE, MoCo-v3, CLIP, and DINO. For a fair comparison, we use the same encoder architecture (ViT-B with a patch size of 16) and ensure that all $\boldsymbol{z}_l$ representations have the same feature dimension. Our results show that image generation quality consistently improves with better visual representations, as measured by KNN accuracy on the ImageNet-1k dataset with $K = 20$.

Features	FID↓	KNN Acc. ↑	
		Top1	Top5
None	14.98	-	-
MAE [98]	10.96	27.44	45.33
MoCo-v3 [45]	10.59	66.57	83.09
CLIP [195]	**6.97**	73.35	**91.12**
DINO [34]	**6.78**	**75.86**	**91.17**

Table 5.5: **Results on COCO text-to-image generation with different visual representations.** We compare the generation quality of a 3-level nested diffusion model, where $L = 3$ and $\{\sigma_l\} = 0.5$, using various visual encoders to construct $\boldsymbol{z}_l$. We report the results *without* CFG. Additionally, we report the accuracy of a KNN classifier with $K = 20$ on ImageNet-1K to quantify feature quality. Our results indicate that better feature quality improves generation results.

5.5 Conclusion

We introduce nested diffusion models, a novel hierarchical generative framework utilizing a succession of diffusion models to generate images starting from low-dimensional semantic feature embeddings and proceeding to detailed image refinement. Unlike conventional single-level latent models and hierarchical models that use low-level feature pyramids, each level in our model is conditional on a more abstract semantic feature hierarchy. This distinctive design improves image structure preservation and maintains global consistency, enhancing generation quality with minimal extra computational expense. We showcase the scalability of our method through a deeper unconditional system, which significantly surpasses the performance of a conditional generation baseline.

CHAPTER 6

LATENT INTRINSICS EMERGE FROM TRAINING TO RELIGHT

Image relighting is the task of showing what a scene from a source image would look like if illuminated differently. Inverse graphics schemes recover an explicit representation of geometry and a set of chosen intrinsics, then relight with some form of renderer. However error control for inverse graphics is difficult, and inverse graphics methods can represent only the effects of the chosen intrinsics. This paper describes a relighting method that is entirely data-driven, where intrinsics and lighting are each represented as latent variables. Our approach produces SOTA relightings of real scenes, as measured by standard metrics. We show that albedo can be recovered from our latent intrinsics without using any example albedos, and that the albedos recovered are competitive with SOTA methods.

6.1 Introduction

Relighting – taking an image of a scene, then adjusting it so it looks as though it had been taken under another light – has a range of applications, including commercial art (e.g., photo enhancement) and data augmentation (e.g., making vision models robust to varying illumination). As a technical problem, relighting is very hard indeed, likely because how a scene changes in appearance when the light is changed can depend on complex surface details (grooves in screws; bark on trees; wood grain) that are hard to capture either in geometric or surface models.

One common approach to relighting a scene is to infer scene characteristics (geometry, surface properties) using inverse graphics methods, then render the scene with a new light source. This approach is fraught with difficulties, including the challenge of selecting which material properties to infer and managing error propagation. These methods perform best in outdoor scenes with significant shadow movements but struggle with indoor scenes where interreflections create complex effects (Section 6.4.2).

As this paper demonstrates, a purely data-driven method offers an attractive alternative. A source scene, represented by an image, is encoded to produce a latent representation of intrinsic scene properties. A source illumination, represented by another image, is encoded to produce a latent representation of illumination properties. These intrinsic and extrinsic properties are combined and then decoded to produce the relighted image. As a byproduct of this training, we find that the latent representation of intrinsic scene properties behaves like an albedo, while another latent representation acts as a lighting controller.

Our model can capture complex scene characteristics without explicit supervision by capturing intrinsic properties as latent phenomena, making it particularly appealing. In contrast to a physical model, we are not required to choose which effects to capture. This latent approach reduces the need for detailed geometric and surface models, simplifies the learning process, and enhances the model's ability to generalize to diverse and unseen scenes. This makes it highly applicable to a wide range of real-world scenarios.

Contributions: We present the first fully data-driven relighting method applicable to images of real complex scenes. Our approach requires no explicit lighting supervision, learning to relight using paired images alone. We demonstrate that this method effectively trains and generalizes, producing highly accurate relightings. Furthermore, we demonstrate that albedo-like maps can be generated from the model without supervision or prior knowledge of albedo-like images. These intrinsic properties emerge naturally within the model. We validate our model on a held-out dataset, applying target lighting conditions from various scenes to assess its generalization capability and precision in real-world scenarios (Section 6.4.2).

6.2 Related Work

Intrinsic Images. Humans have been known to perceive scene properties independent of lighting since at least 1867 [250, 100, 16, 81]. In computer vision, the idea dates to Barrow and Tenenbaum [14] and comprises at least depth, normal, albedo, and surface material maps. Depth and normal estimation are now well established (eg [123]). There is a rich literature on albedo estimation (dating to 1959 [147, 148]!). A detailed review appears in [75], which breaks out methods as to what kinds of training data they see. Early methods do not see any form of training data, but more recently both CGI data and manual annotations of relative lightness (labels) have become available. Early efforts, such as SIRFS [12], focused on using shading information to recover shape, illumination, and reflectance, highlighting the importance of modeling these factors for intrinsic image analysis. Recent strategies include: deep networks trained on synthetic data [159, 116, 72]; and conditional generative models [139].

The weighted human disagreement ratio (WHDR) evaluation framework was introduced by [17] using the IIW dataset. This is a dataset of human judgments that compare the absolute lightness at pairs of points in real images. Each pair is labeled with one of three cases (first lighter; second lighter; indistinguishable) and a weight, which captures the certainty of labelers. One evaluates by computing a weighted comparison of algorithm predictions with human predictions; WHDR scores can be improved by postprocessing because most methods produce albedo fields with very slow gradients, rather than piecewise constant albedos. [25] demonstrate the value of "flattening" albedo (see also [182]); [26] employ a fast bilateral filter [13] to obtain significant improvements in WHDR.

Using Intrinsic Images for Relighting. Bhattad and Forsyth [21] demonstrated that intrinsic images could be used for reshading inserted objects. This approach can be extended by adjusting the shading in both the foreground and background to eliminate discrepancies [31]. Intrinsic images and geometry-aware networks have been used for multi-view relighting [192]. StyLitGAN [24] introduced a method to relight images by identifying directional vectors in the latent space of StyleGAN, but can only relight StyleGAN generated images and requires explicit albedo and shading to guide relighting. It can be extended to real images using a GAN inversion, but does not generalize [22]. LightIt [138] controls lighting changes in image generation using diffusion models, by conditioning on shading and normal maps to achieve consistent and controllable lighting. Like these methods, we use intrinsics and extrinsics to relight, but ours are latent, with no explicit physical meaning.

Color Constancy. Image color is ambiguous: a green pixel could be the result of a white light on a green surface, or a green light on a white surface. Humans are unaffected by this ambiguity (eg [100, 16]; recent review in [263]). There is extensive computer vision literature; a recent review appears in [153]. We do not estimate illumination color but estimate a single color correction (Section 6.4.2).

Lighting Estimation and Representation. Accurate lighting representation is crucial for tasks like object insertion and relighting. Traditional methods used parametric models such as environment maps and spherical harmonics to represent illumination [57, 198]. Debevec's seminal work [57] on recovering environment maps from images of mirrored spheres set the foundation for many subsequent works. Methods by Karsch et al. [132, 133], Gardner et al. [76, 77], Garon et al. [78] and Weber at al. [261] advanced the field by using learned models to recover parametric, semi-parametric or panoramic representations of illumination. Recent approaches include representing illumination fields as dense 2D grids of spherical harmonic sources [160, 162] or learning 3D volumes of spherical Gaussians [260]. These methods can model complex light-dependent effects but require extensive CGI datasets for training [203, 161]. Our approach diverges by not relying on labeled illumination representations or CGI data, instead producing abstract representations of illumination through deep features without specific physical interpretations.

Image-based Relighting. Other works focus on portrait relighting using deep learning [233, 295, 183, 218], which are typically specialized to faces and trained on paired or light-stage data. Self-supervised methods for outdoor image relighting leverage single-image decomposition with parametric outdoor illumination, benefiting from simpler lighting conditions dominated by sky and sunlight [281, 165]. [109] introduced a self-attention autoencoder model to re-render a source image to match the illumination of a guide image, focusing on separating scene representation and lighting estimation with a self-attention mechanism for targeted relighting. Similarly, [271] proposed a depth-guided image relighting, which combines source and guide images along with their depth maps to generate relit images. In contrast, our work shows that intrinsic properties relevant to relighting can emerge naturally from training to relight, facilitating complex scene relighting without the need for explicit lighting estimation. We compare with both [109] and [271] for relighting capabilities on real scenes.

Emergent Intrinsic Properties. Bhattad et al. [23] and Du et al. [67] demonstrate that intrinsic images can be extracted from generative models using a small intrinsic image dataset obtained from pretrained off-the-shelf intrinsic image models. Our work explores how intrinsic image properties emerge as a result of training a model for relighting, without the need for an intrinsic image dataset.

6.3 Learning Latent Intrinsic from Relighting.

Our relighting model can be seen as a form of autoencoder. One encoder computes a latent representation of scene intrinsics from an image of a target scene; another computes a latent representation of scene extrinsics from an image of a placeholder scene in the reference lighting. These are combined, then decoded into a final image of the target scene in the reference lighting. Losses impose the requirements that (a) the final image is right and (b) the latent intrinsics computed for a scene are not affected by illumination. The procedure for combining intrinsics and extrinsics is carefully designed to make it very difficult for intrinsic features of the placeholder scene to "leak" into the final image.

6.3.1 Model structure

Encoder setup: Write $\boldsymbol{I}_s^l \in \mathrm{R}^{H \times W \times 3}$ for the input image, captured from scene s with lighting configuration l. Training uses pairs $\boldsymbol{I}_s^{l_1}$ and $\boldsymbol{I}_s^{l_2}$, representing the same scene s under different lighting conditions l_1 and l_2. The model *does not see* detailed lighting information (for example, the index of the lighting) during training, because standardizing lighting settings across various scenes is often impractical.

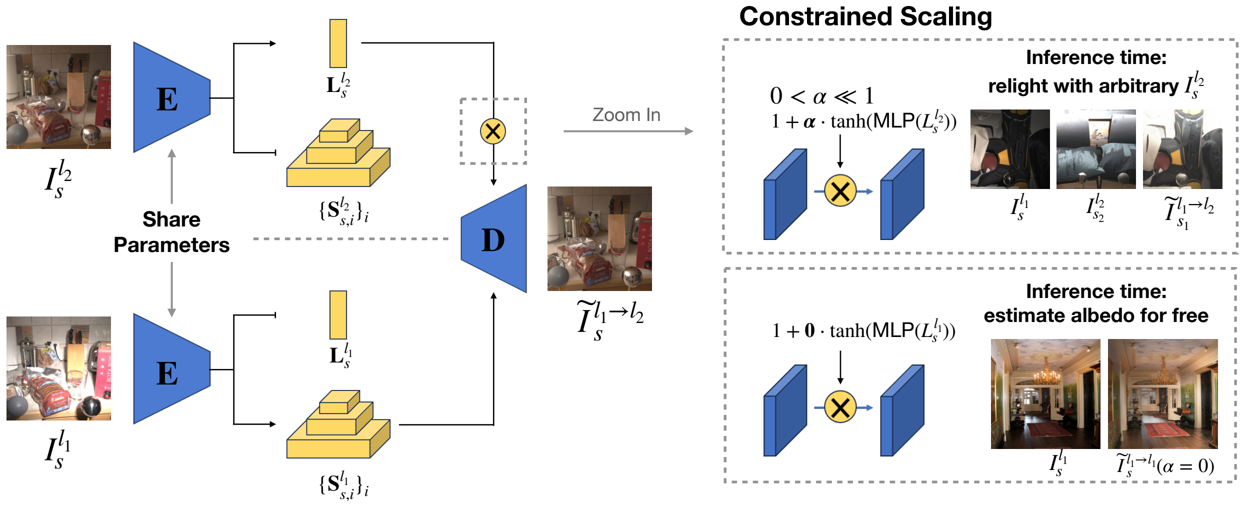

Constrained Scaling
E
E
D
$\mathbf{L}_s^{l_2}$
$\{\mathbf{S}_{s,i}^{l_2}\}_i$
$\mathbf{L}_s^{l_1}$
$\{\mathbf{S}_{s,i}^{l_1}\}_i$
$I_s^{l_2}$
$I_s^{l_1}$
Share Parameters
$\widetilde{I}_s^{l_1 \to l_2}$
Zoom In
Inference time: relight with arbitrary $I_s^{l_2}$
$0 < \alpha \ll 1$
$1 + \boldsymbol{\alpha} \cdot \tanh(\mathrm{MLP}(L_s^{l_2}))$
$I_s^{l_1}$
$I_{s_2}^{l_2}$
$\widetilde{I}_{s_1}^{l_1 \to l_2}$
Inference time: estimate albedo for free
$1 + \mathbf{0} \cdot \tanh(\mathrm{MLP}(L_s^{l_1}))$
$I_s^{l_1}$
$\widetilde{I}_s^{l_1 \to l_1}(\alpha = 0)$

Figure 6.1: The network diagram of our relighting model. The model functions as an autoencoder, comprising an encoder $\boldsymbol{E}$ and a decoder $\boldsymbol{D}$. ***Left Half***: The encoder $\boldsymbol{E}$ maps input image $\boldsymbol{I}_s^l$, captured under scene s and lighting l, to low-dimensional extrinsic features $\boldsymbol{L}_s^l$ and set of intrinsic features map $\{S_{s,i}^l\}_i$. The decoder $\boldsymbol{D}$ then generates new images based on these intrinsic and extrinsic representations. ***Right Half***: We employ *constrained scaling* for the injection of $\boldsymbol{L}_s^l$, utilizing $0 < \alpha \ll 1$ to regularize the information passed from $\boldsymbol{L}_s^l$, thereby enforcing a low-dimensional parameterization of the extrinsic features. We train our system to relight target images given input paired with images captured under the same scene s. During inference, our model demonstrates the ability to generalize to arbitrary reference images for relighting and can estimate albedo for free.

Write E for the encoder, D for the decoder. The encoder must produce the intrinsic and extrinsic representations from the input image. Write $\boldsymbol{S}_{s,i}^l \in \mathbf{R}^{(H_i \times W_i) \times C_i}$ for spatial feature maps yielding the intrinsic representation, with i for the layer index, and $\boldsymbol{L}_s^l \in \mathbf{R}^C$ for extrinsic features; we have:

$$E(\boldsymbol{I}_s^l) := \{\boldsymbol{S}_{s,i}^l\}_i, \boldsymbol{L}_s^l \tag{6.1}$$

We apply L2 normalization along the feature channel to both sets of features. During training, we add random Gaussian noise to the input image to enhance semantic scene understanding capabilities:

$$E(\boldsymbol{I}_s^l + \sigma\epsilon) := \{\boldsymbol{S}_{s,i}^l\}_i, \boldsymbol{L}_s^l \tag{6.2}$$

Decoder setup: The decoder D relights $\boldsymbol{I}_s^{l_1}$ using extrinsic features extracted from $\boldsymbol{I}_s^{l_2}$:

$$D(\{\boldsymbol{S}_s^{l_1}\}, \boldsymbol{L}_s^{l_2}) := \tilde{\boldsymbol{I}}_s^{l_1 \to l_2} \tag{6.3}$$

We optimize the autoencoder using a pixel-wise loss on both relighted and reconstructed images:

$$\mathcal{L}_{\text{relight}} := \mathcal{L}_{\text{pixel}}(\tilde{\boldsymbol{I}}_s^{l_1 \to l_2}, \boldsymbol{I}_s^{l_2}) + \mathcal{L}_{\text{pixel}}(\tilde{\boldsymbol{I}}_s^{l_2 \to l_2}, \boldsymbol{I}_s^{l_2}) \tag{6.4}$$

where $\mathcal{L}_{\text{pixel}}$ represents the pixel-wise losses: L2 distance on pixels; structural similarity index (SSIM) [259]; and l2 distance on image spatial gradient (weights 10, 0.1 and 1 respectively).

6.3.2 Intrinsicness

Intrinsicness: Our model should report the same latent intrinsic for the same scene in different lightings, so we apply the following loss to the encoder:

$$\mathcal{L}_{\text{intrinsic}} := \sum_i \|\boldsymbol{S}_{s,i}^{l_1} - \boldsymbol{S}_{s,i}^{l_2}\|_2 + \text{1e-3} \cdot \mathcal{L}_{\text{reg}}(\boldsymbol{S}_{s,i}^{l_1}) \tag{6.5}$$

where $\mathcal{L}_{\text{reg}}$ is a regularization term on intrinsic features, defined as follows:

$$\begin{aligned} \mathcal{L}_{\text{reg}}(\boldsymbol{S}) &:= \|R(\boldsymbol{S}) - R(\hat{\boldsymbol{S}})\|_2 \quad (6.6) \\ R(\boldsymbol{S}) &:= \log\det\left(\boldsymbol{I} + \frac{d}{n\lambda^2}\boldsymbol{S}^\top \boldsymbol{S}\right) \quad (6.7) \end{aligned}$$

where $R(\boldsymbol{S})$ is the coding rate [279] for a matrix $\boldsymbol{S} \in \mathbb{R}^{n \times d}$ with each row l2 normalized, under a distortion constant λ. $\hat{\boldsymbol{S}}$ is a random matrix with the same shape of $\boldsymbol{S}$ and each row of $\hat{\boldsymbol{S}}$ is sampled from uniform hyperspherical distribution at the start of learning. In Eqn.6.5, $R(\hat{\boldsymbol{S}})$ serves as the optimization target of $R(\boldsymbol{S})$ to encourage the $\boldsymbol{S}$ to uniformly spread out in the hyperspherical space. This strategy is now widely used in self-supervised learning; without the regularization term, the model can minimize the feature distance by simply collapsing the distribution of $\boldsymbol{S}^l_{s,i}$ with small variance, which will not yield effective lighting invariance.

6.3.3 *Combining intrinsics and extrinsics*

The placeholder scene is necessary to communicate illumination to the model, but has important nuisance features. Intrinsic information from this scene could "leak" into the final image, spoiling results. We introduce *constrained scaling*, a structural bottleneck that restricts the amount of information transmitted from the learned extrinsic features.

Write $\boldsymbol{F} \in \mathbf{R}^{h\times w\times c}$ for the feature map fed to the decoder. Constrained scaling combines intrinsic and extrinsic features by

$$\tilde{\boldsymbol{F}} := \boldsymbol{F} \odot \left(1 + \alpha \cdot \tanh\left(\text{MLP}\left(\boldsymbol{L}^l_s\right)\right)\right) \tag{6.8}$$

where MLP, a series of fully connected layers with non-linear activation, aligns $\boldsymbol{L}^l_s$ to the latent channel dimension of $\boldsymbol{F}$ and $\alpha \ll 1$ is a small non-negative scalar (we use 5e-3). This approach means that any single extrinsic feature vector has little effect on the feature – for an effect, the extrinsics must be pooled over multiple locations. Illumination fields tend to be spatially smooth, supporting the insight that enforced pooling is a good idea.

Constrained scaling compresses latent vectors into a very small numerical range, making learning difficult. We use a regularizer to promote a uniform distribution of $\boldsymbol{L}^l_s$, which improves optimization. In particular, we have

$$\mathcal{L}_{\text{extrinsic}} := \mathcal{L}_{\text{reg}}(\boldsymbol{L}^l_s) \tag{6.9}$$

By choosing $\alpha \ll 1$ and training model with uniform regularization term Eqn.6.9, we effectively push the lighting code to uniformly spread over $[-\alpha, \alpha]$ where the absolute value of each channel indicates the strength of the light. As a side effect, by setting $\alpha = 0$ to disable the contribution of the lighting code, we get image albedo estimation from our model for free.

Our final training objective is weighted combination of all individual loss terms:

$$\mathcal{L} := \mathcal{L}_{\text{relight}} + \text{1e-1} \cdot \mathcal{L}_{\text{intrinsic}} + \text{1e-4} \cdot \mathcal{L}_{\text{extrinsic}} \tag{6.10}$$

6.4 Experiments

We will first provide a brief description of our experimental procedure (Sec 6.4.1), followed by a discussion on how we evaluate the various relighting capabilities of our approach, including its strong generalization across datasets with different distributions (Sec 6.4.2). Finally, we will present the emergent albedo that is recovered from the latent intrinsic without using any albedo-like images (Sec 6.4.3).

6.4.1 Experiment Details

Training details: We train our model using the MIT multi-illumination dataset [181], which includes images of 1,015 indoor scenes captured under 25 fixed lighting, totaling 25,375 images. We follow the official data split and train our model on the 985 training scenes. During training, we randomly sample pairs of images from the same scene under different lighting conditions and perform random spatial cropping, with the crop ratio randomly selected between 0.2 and 1.0, followed by resizing the cropped image to a resolution of 256x256. For further details, please refer to our appendix.

6.4.2 Evaluating image relighting

Relighting on the Multi-illumination dataset: We relight images of scenes in the test set using reference images from the test set, then compare to the correct known relighting from the test set using various metrics. For each input image, we randomly sample reference images from different scenes and lighting conditions. To reduce the effect of randomness in comparing different

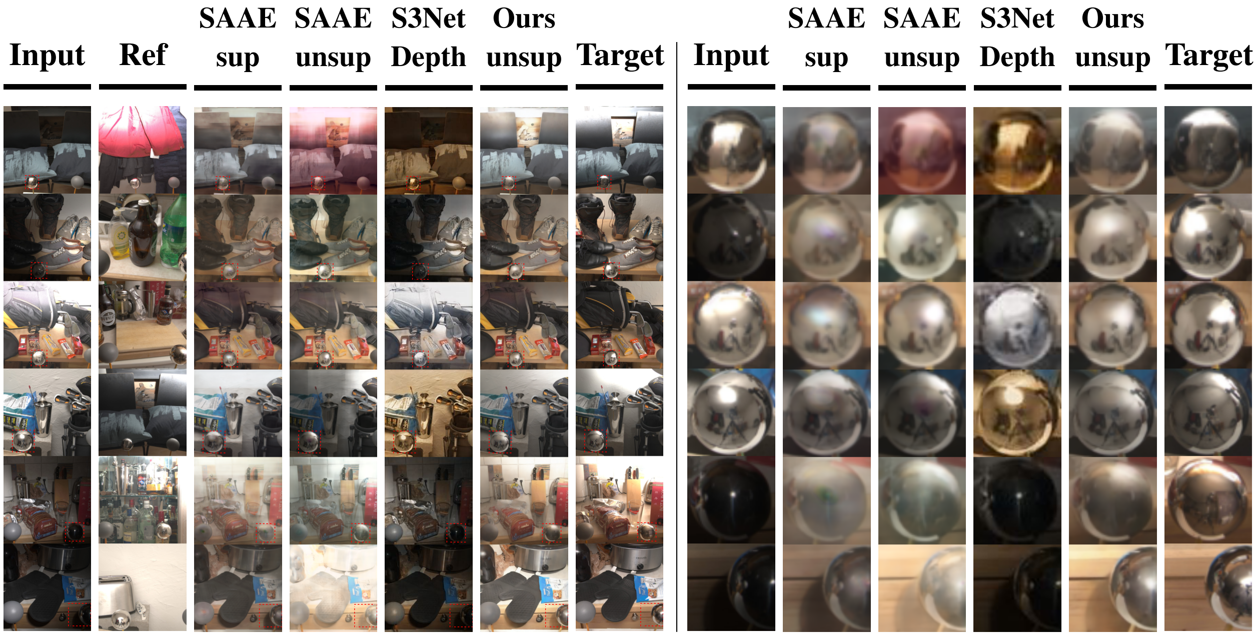


Figure 6.2: Our method outperforms all other approaches in estimating light and rendering the scene. The Unsupervised SA-AE [109] method fails by incorporating intrinsic elements from reference images. The S3Net [271] approach struggles with rendering when using unpaired reference images. ***Right***: A zoomed-in view of the chrome ball was used as a probe to evaluate detail preservation in the environment map. Our method effectively retains the intricate room layout and accurately renders the appropriate lighting effects.

relighting strategies, we select 12 random reference images for each input image, and maintain the same image-reference pairs when evaluating different models. We report the results, measured in RMSE and SSIM, in Table 6.1. We report these metrics both for absolute predictions and for predictions where any global color shift is corrected by a single, least-squares scale of each predicted color layer (i.e. one scale for R; one for G; one for B). This color correction allows us to distinguish between spatial errors and global color shifts; these appear to have a significant effect, possibly because there are visible color shifts present in some of the dataset images.

In Table 6.1, we compare to SA-AE [109], a model that requires a ground truth light index for supervision, and S3-Net, which needs a ground truth depth map as a conditional input. For S3-Net, we use a state-of-the-art depth estimator to provide pseudo-GT on the relighting dataset as input. For a fair comparison to our model, which does not require any supervision outside of the ground truth relighting, we also report results for modified versions of the baselines trained without

Methods	**Labels**	**Raw Output**		**Color Correction**	
		RMSE↓	SSIM↑	RMSE↓	SSIM↑
Input Img	-	0.384	0.438	0.312	0.492
SA-AE [109]	Light	**0.288**	**0.484**	0.232	0.559
SA-AE [109]	-	0.443	0.300	0.317	0.431
S3Net [271]	Depth	0.512	0.331	0.418	0.374
S3Net [271]	-	0.499	0.336	0.414	0.377
Ours($\sigma = 0$)	-	0.326	0.232	0.242	0.541
Ours(w/o $\mathcal{L}_{reg}$)	-	0.315	0.462	0.232	0.550
Ours	-	0.297	0.473	**0.222**	**0.571**

Table 6.1: We assess the quality of image relighting using the multi-illumination dataset [181]. Our method, when evaluated on raw output, significantly outperforms all other unsupervised approaches and achieves competitive results compared to the supervised SA-SA [109], which requires ground truth light supervision. When we correct the colors by eliminating global color drift caused by light ambiguity, our method surpasses all other approaches. Additionally, warming up the model as a denoising autoencoder proves beneficial compared to when it is not warmed up ($\sigma = 0$).

α	**Raw Output**		**Color Correction**	
	RMSE ↓	SSIM ↑	RMSE↓	SSIM↑
∞	0.471	0.287	0.352	0.407
1e-2	0.314	0.444	0.238	0.546
5e-3	**0.297**	**0.473**	**0.222**	**0.571**
1e-3	0.312	0.453	0.256	0.524
5e-4	0.309	0.460	0.253	0.533

Table 6.2: We analyze the impact of α on relighting quality using the multi-illumination dataset [181]. Setting α to ∞, which removes the scaling constraints, results in poor relighting quality, indicating that restricting information from extrinsic sources significantly improves generation quality. Within a limited parameter search, 5e-3 yields the best results.

additional supervision. For SA-AE, we train their light estimation model and relighting model end-to-end by removing the loss from light supervision. For S3-Net [271], we simply remove the depth from the model's input.

Without color correction, only light-supervised SA-AE slightly outperforms our model, while all other baselines are significantly worse. The unsupervised version of SA-AE performs much worse because their light estimator struggles to distinguish the extrinsic from the intrinsic components. Specifically, SA-AE also parameterizes the extrinsic as a lower-dimensional representation

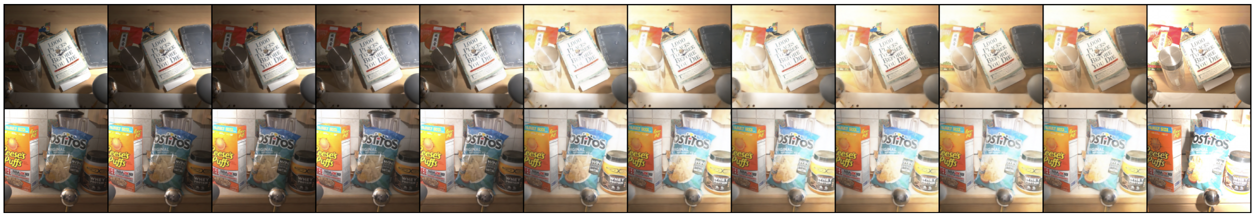


Figure 6.3: Latent extrinsics can be interpolated successfully; **leftmost** and **rightmost** columns are images from the multi-illumination dataset, and intermediate images are obtained by linear interpolation on the latent extrinsics (light-dependent representations), then decoding. Note how the light seems to "move" across space.

but without the constrained scaling that our model uses. As a result, the estimated extrinsic from their unsupervised model also carries intrinsic information, and one can see “leaks”. S3-Net performs worse in both versions since they concatenate input and reference images before feeding them into the models, which significantly affects the model’s generalization ability, especially during test time when we use images from different scenes as references.

On color-corrected images, our approach outperforms all methods, including the light-supervised version of SA-AE, indicating that, up to the constant color drift, our extrinsic estimation network is at least as good as, or even better than, a light estimation network trained with supervision. Removing the denoising setup from our model ($\sigma = 0$) results in worse performance in both cases due to inferior semantic scene understanding. We additionally provide ablation studies on the choices of α in Table 6.2 and find $\alpha = 5e - 3$ produces the best results.

Each image in the multi-illumination dataset shows a chrome ball, which gives a good estimate of an environment map for that image. Correctly rendering the effects of lighting changes on these chrome balls appears to be extremely difficult; the changes are substantial, and concentrated in a small region of the image (so correct representation of these changes has little effect on typical image losses). Figure 6.2 shows a crop of our results around this chrome ball. Our method represents these changes well; we are aware of no other results reported for this effect. Compared to other approaches, our model accurately preserves the room layout, even in cases of extreme light changes.

Unlike classical rendering models that use a specific parameterized form to represent extrinsics, our framework learns an implicit extrinsic representation. However, we can still parameterize the learned extrinsic representation to create new light sources. In Figure 6.3, we demonstrate this capability by rendering images using interpolated extrinsic representations.

Relighting synthetically relighted images from StyLitGAN: StyLitGAN [24] is a recent method that can produce multiple illuminations of a single generated room scene by manipulating StyleGAN latents appropriately. In the multi-illumination dataset, reference light and target

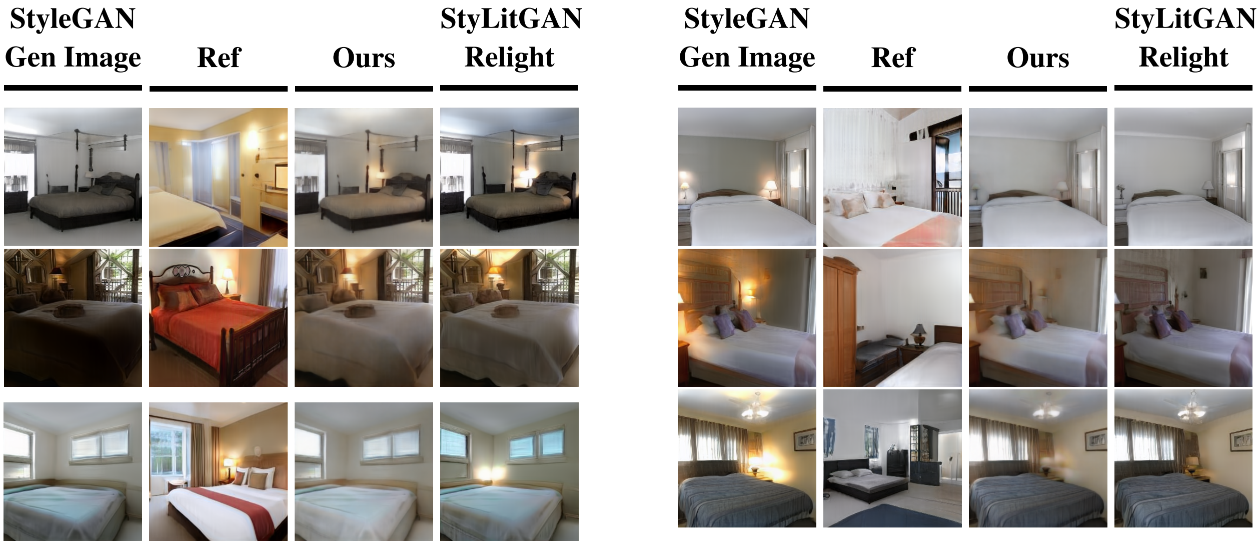


Figure 6.4: Qualitative results for relighting interior scenes using our relighter trained on images obtained from StyLitGAN (which produces multiple illuminations of a generated scene). StyLitGAN has a strong tendency to increase or decrease illumination by adjusting luminaires, typically bedside lights but also light coming through French windows, etc. On the **left**, where the reference lighting tends to be brighter and more concentrated, notice how for the two top images, our relighter has identified and "turned up" the bedside lights; for the third, it has resisted StyLitGAN's tendency to invent helpful luminaires (there isn't a bedside light where StyLitGAN imputed one, as close inspection shows). On the **right**, where the reference lighting is much more uniform, our relighter has achieved this by "turning down" bedside lights. This is an emergent phenomenon; the method is not supplied with any explicit luminaire model or labeled data.

images tend to share a strong spatial correlation in light patterns. In contrast, StyLitGAN generates extremely challenging images where very significant changes in lighting occur. Furthermore, StyLitGAN images have visible luminaires. To relight the input, the model must infer high-level concepts rather than simply copying the spatially corresponding light patterns from the reference. We train our model using StyLitGAN images to evaluate generalization qualitatively (quantitative evaluation would be of dubious value, because StyLitGAN images are generated rather than real). Figure 6.4 shows results. Notice how our method successfully relights from references, achieves brighter illuminations by turning on luminaires (here bedside lights), achieves darker scenes by turning off luminaires, and is somewhat less inclined to invent luminaires than StyLitGAN is. The model knows that light must come from somewhere, and how the effects of light are distributed.

Zero-Shot Relighting: In Figure.6.5, we show our model's strong generalization by apply-

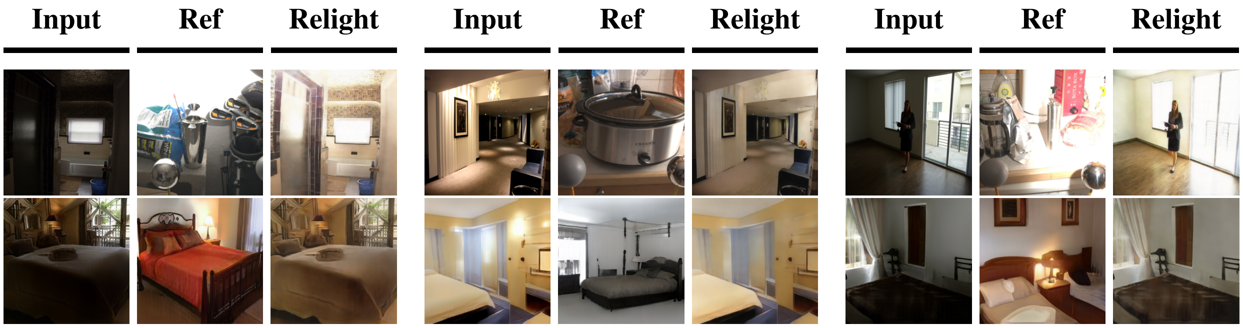


Figure 6.5: **Zero-Shot Relighting.** Our relighting model, trained only on the multi-illumination dataset, generalizes well to out-of-distribution images, as shown on the IIW dataset (first row) and StyleGAN images (second row). It accurately infers scene geometry and lighting. Note that it identifies and turns on the bedside lamps in StyleGAN images despite having no training in bedroom images. This demonstrates the model's strong generalization ability and the model clearly “knows" something about light sources.

ing the model solely trained on multi-illumination dataset—without additional training or fine-tuning—to relight IIW and StyleGAN-generated images. Despite the significant distribution shift in lighting patterns and room setup, our model accurately identifies luminaires and relights images.

6.4.3 Zero-shot albedo evaluation

Constrained scaling allows us to infer albedo without any decoding (and without any albedo data!) by setting $\alpha = 0$ during inference. We benchmark these albedo estimates using the WHDR metric on the IIW [17] dataset (Section 6.2). We use WHDR because it is widely used and allows comparisons, but existing literature records significant problems in interpreting the measure [75, 21, 139]. Among other irritating features, the metric seems to prefer odd colors, and can be hacked by heavily quantized albedo maps. As is standard, we obtain lightness by averaging R, G, and B albedo and compute relative lightness of two pixel locations i_1, i_2 by comparing to a confidence threshold δ:

$$\widetilde{J}_{i,\delta}(\bar{R}) = \left\{ \begin{array}{cc} 1 & \text{if } \bar{R}_{i_1}/\bar{R}_{i_2} > 1+\delta \\ 2 & \text{if } \bar{R}_{i_2}/\bar{R}_{i_1} > 1+\delta \\ E & \text{otherwise} \end{array} \right\} \tag{6.11}$$

Methods	labels	Flat	Tune δ	WHDR
Intrinsic Diffusion [139]	CG	No	No	22.61
Intrinsic Diffusion[139]	CG	Yes	Yes	17.10
Inverser Render[280]	No	No	No	21.40
BBA[75]	No	No	Yes	17.04
Ours	No	No	No	28.97
Ours	No	No	Yes	19.09
Ours	No	Yes	Yes	**15.81**

Table 6.3: We benchmark our albedo esimation on test set of IIW dataset [17] and compare with others, though the reliability has been questioned by recent papers [75]. Flat denotes postprocessing images with flattening [25]. Despite our model never being trained on albedo maps or CG data, our best configuration significantly outperforms all other methods suggesting our model learns high-quality intrinsic representations

α	WHDR			
	δ = 0.1		optimal δ	
	w/ F	w/o F	w/ F	w/o F
1e-2	**17.64**	**28.97**	**15.81**	**19.09**
5e-3	18.93	31.81	16.02	19.53
1e-3	18.00	29.77	15.84	19.13
5e-4	18.04	29.62	15.85	19.12

Table 6.4: We conduct ablation experiments to assess the impact of α on the quality of albedo. "w/F" and "w/o F" denote post-processing images with and without flattening [25], respectively. The setting of $\delta = 0.1$ and w/o F is the most affected by α. Despite this, all values of α achieve high performance in our optimal configurations.

The resulting classification (one lighter than two; two lighter than one; equivalent) is then compared to human annotations J using the confidence score w_i for each annotation pair. We report WHDR on the IIW test split in Table 6.3 to facilitate comparison with other approaches. Since our model is not trained with any albedo maps or computer-generated images, we need to adjust the threshold for the optimal performance. Following prior work, we optimize δ on the training split, which significantly improves our performance from 28.97 to 19.09. Additionally, we enhance our performance by post-processing our albedo map using flattening [25], an optimization technique to further reduce color variations. With this improvement, our results reach 15.81, substantially outperforming the intrinsic diffusion model [139], a diffusion-based albedo regression model trained on computer graphics data. In Figure 6.6, we show some qualitative comparisons to intrinsic diffusion. We observe that our method effectively removes external lighting effects and does not suffer from color drift due to domain gap unlike intrinsic diffusion, which is trained on CG data.

Sensitivity to light changes: Albedo are scene properties that are independent of lighting changes. In Figure. 6.7, we qualitatively assess this characteristic by varying lighting conditions,

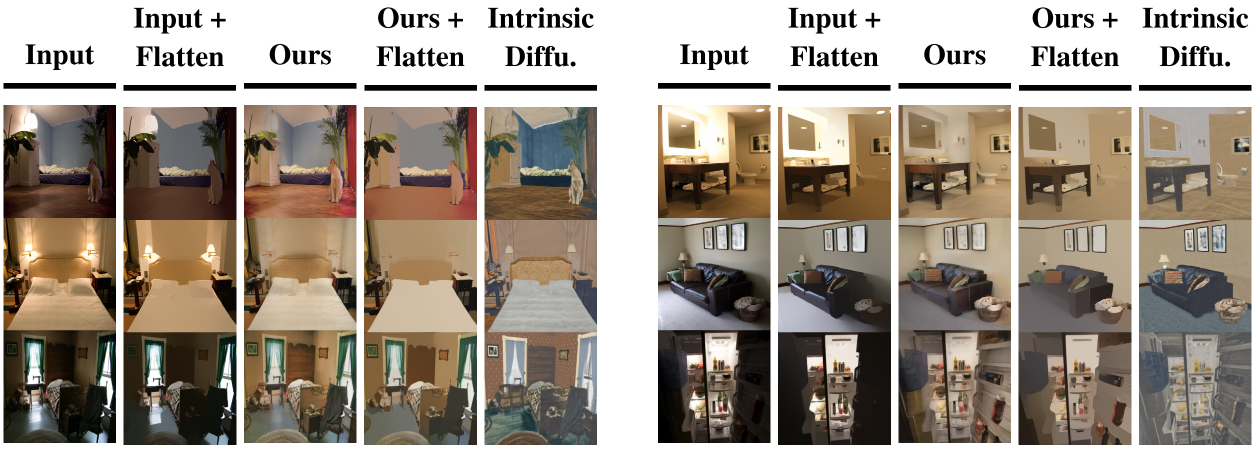


Figure 6.6: Qualitative Comparison of **Emergent Albedo from Latent Intrinsics** on the IIW Dataset. Although our model has never been trained on any albedo-like maps, it effectively removes the effects of external light and dark shadows from the input. In contrast, Intrinsic Diffusion [139], a supervised method trained on large computer graphics data, often produces color-drifted estimations, likely due to the domain shift between CG data and real images. Observe the subdued lighting around the mirrors (top row, right) in our recovered albedo. Also, pay attention to all the details inside the refrigerator, which are visible in our recovered albedos (bottom row; right) compared to intrinsic diffusion. For comparison, we also display naive flattening (in the second column), which by itself cannot effectively reduce the strong lighting effects.

comparing our approach with the state-of-the-art supervised method, Intrinsic Diffusion [139]. Our method demonstrates consistent and accurate estimations that remain stable even under extreme lighting variations. In contrast, Intrinsic Diffusion [139] shows significant deviation from the natural color distribution and are sensitive to lighting changes.

6.5 Discussion, Limitations and Future Work

Our method presents an important advancement in image relighting by demonstrating that intrinsic properties such as albedo can emerge naturally from training on relighting tasks without explicit supervision. This finding simplifies the relighting process, eliminating the need for detailed geometric and surface models and enhancing the model's ability to generalize across diverse and unseen scenes. By encoding scene and illumination properties as latent variables, we achieve accurate and flexible relighting. Our findings will have implications for various fields such as virtual reality and cinematic post-production. This approach reduces the learning process's complexity

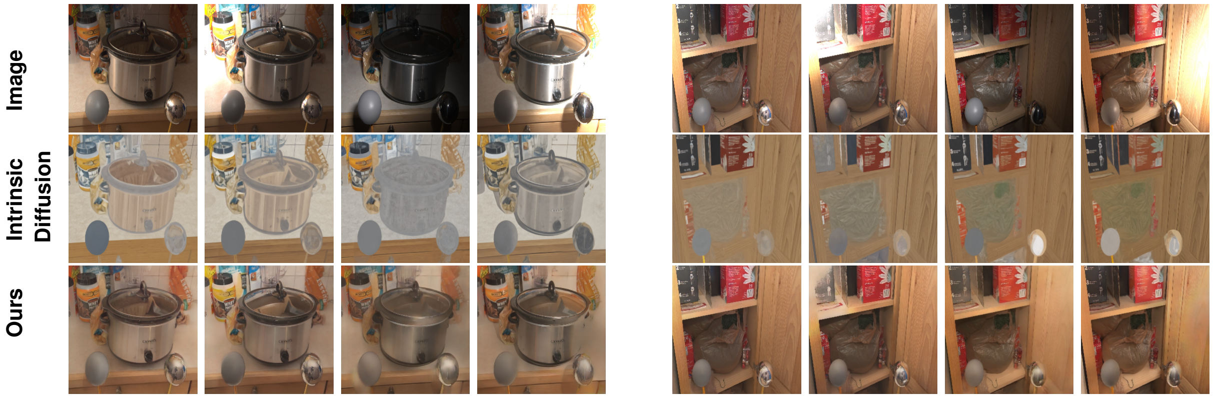


Figure 6.7: Qualitative comparison of albedo stability under varying lighting conditions. Images shown are from the multi-illumination dataset test split. The top row features images under different lighting environments. The middle row presents estimated albedos obtained from Intrinsic Diffusion [139], while the bottom row shows the recovered albedos from the latent intrinsic representation. Intrinsic Diffusion has large color drift and is sensitive to changes in lighting. In contrast, the **albedos recovered from latent intrinsics remain stable under lighting changes, even in extreme conditions.**

and offers a new perspective on designing deep learning models to capture and utilize intrinsic scene properties. These findings can guide future research toward a more efficient and scalable relighting approach, encouraging the development of models that can handle various lighting conditions and scene complexities.

The current taxonomy of surface intrinsics—typically, depth, normal, albedo, and perhaps specular albedo and roughness—is quite limiting (compare human language for surface properties [16]). Our method, which computes latent intrinsic and extrinsic representations from images and combines these to transfer lighting conditions across scenes, captures physical concepts like luminaire and albedo without explicit physical parametrization. This ability to represent significant image effects without choosing a surface model offers substantial flexibility.

However, our method has several limitations. It relies on pairs of relighted data captured in the same scene, which can be resource-intensive to obtain. Additionally, it does not cope well with saturated pixel values common in LDR images. The intrinsic information being latent is another limitation since many applications require explicit intrinsic information like depth and normals.

Nonetheless, there is good evidence that explicit intrinsic information can be extracted from our latent intrinsics. Our method clearly “knows” albedo, and this information can be elicited without examples. Similarly, it “knows” something about luminaires, such as their locations and effects. It is intriguing to speculate that it “knows” other information relevant to relighting, such as depth or surface microstructure. Future work will pursue this line of inquiry and also focus on developing a purely unsupervised framework to infer intrinsic and extrinsic properties from collections of in-the-wild images. This will include refining probing techniques for better extraction of explicit intrinsics and identifying additional intrinsic properties crucial for relighting that do not align with the current taxonomy. We believe this will improve the applicability and robustness of our approach, making it suitable for a wider range of real-world scenarios.

CHAPTER 7

CONCLUSION

This thesis explores the bidirectional relationship between representation learning and generative models, demonstrating how advances in one can reinforce progress in the other. The first part focuses on improving representation learning within generative models through decayed identity shortcuts, structure-aware adversarial objectives, and scalable clustering techniques for inferring low-dimensional embeddings. These methods collectively push generative models toward more effective unsupervised representation learning.

In the second part, we study the reverse perspective: we show how incorporating structured representations can significantly enhance generation quality. We introduce a hierarchical generation framework that operates in a semantically meaningful latent space, achieving both higher compression and improved structural fidelity in the generated outputs. Additionally, we propose a generative framework for relighting that leverages physically-grounded regularization to disentangle intrinsic and extrinsic image features, improving the interpretability and performance of the relighting task.

Although this thesis investigates the mutual enhancement of generative modeling and representation learning, several promising directions remain open. One key challenge is integrating visual representation learning and a hierarchical generation model within a unified, jointly trained framework. Rather than relying on pre-trained encoders, future systems could learn to both understand and synthesize data in a self-supervised loop, enabling tighter coupling between perception and generation.

Another important direction is to extend representation learning and generative modeling to multimodal settings, such as integrating vision with language or audio. This involves jointly learning to embed data from different modalities into a shared latent space, and training generative models to map samples from that space back into each domain. Such a framework could significantly improve the controllability and semantic alignment of the generation process.

Lastly, grounding generative models in structured priors—motivated by physics or geometry—offers new opportunities for building interpretable, controllable, and generalizable systems. Understanding how these priors interact with learned representations remains an open and promising area for future exploration.

REFERENCES

[1] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *TOG*, 2021.

[2] Korbinian Abstreiter, Sarthak Mittal, Stefan Bauer, Bernhard Schölkopf, and Arash Mehrjou. Diffusion-based representation learning. *arXiv:2105.14257*, 2021.

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv:2303.08774*, 2023.

[4] Pablo Arbeláez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *PAMI*, 2011.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, 2017.

[6] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *NeurIPS*, 2019.

[7] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, 2023.

[8] 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. In *ICML*, 2022.

[9] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A ViT backbone for diffusion models. In *CVPR*, 2023.

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

[11] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *ICLR*, 2021.

[12] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *PAMI*, 2014.

[13] Jonathan T. Barron and Ben Poole. The fast bilateral solver. In *ECCV*, 2016.

[14] H.G. Barrow and J.M. Tenenbaum. Recovering intrinsic scene characteristics from images. In *ICVS*, 1978.

[15] Daniel Beaglehole, Adityanarayanan Radhakrishnan, Parthe Pandit, and Mikhail Belkin. Mechanism of feature learning in convolutional neural networks. *arXiv:2309.00570*, 2023.

[16] J. Beck. *Surface color perception.* Cornell University Press, 1972.

[17] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic Images in the Wild. In *SIGGRAPH*, 2014.

[18] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *TPAMI*, 2013.

[19] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision. In *ICCV*, 2015.

[20] Vineeth S Bhaskara, Tristan Aumentado-Armstrong, Allan D Jepson, and Alex Levinshtein. Gran-gan: Piecewise gradient normalization for generative adversarial networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.

[21] Anand Bhattad and David A Forsyth. Cut-and-paste object insertion by enabling deep image prior for reshading. In *3DV*, 2022.

[22] Anand Bhattad, Viraj Shah, Derek Hoiem, and David A Forsyth. Make it so: Steering stylegan for any image inversion and editing. *arXiv:2304.14403*, 2023.

[23] Anand Bhattad, Daniel McKee, Derek Hoiem, and David Forsyth. Stylegan knows normal, depth, albedo, and more. In *NeurIPS*, 2024.

[24] Anand Bhattad, James Soole, and D.A. Forsyth. Stylitgan: Image-based relighting via latent control. In *CVPR*, 2024.

[25] Sai Bi, Xiaoguang Han, and Yizhou Yu. An l 1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. In *SIGGRAPH*, 2015.

[26] Sai Bi, Nima Khademi Kalantari, and Ravi Ramamoorthi. Deep hybrid real and synthetic training for intrinsic decomposition. In *EGSR*, 2018.

[27] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv:2108.07258*, 2021.

[28] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning epresentations*, 2019.

[29] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018.

[30] John Canny. A computational approach to edge detection. *PAMI*, 1986.

[31] Chris Careaga, S Mahdi H Miangoleh, and Yağız Aksoy. Intrinsic harmonization for illumination-aware image compositing. In *SIGGRAPH Asia*, 2023.

[32] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[33] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 2020.

[34] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[35] Arantxa Casanova, Marlene Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero Soriano. Instance-conditioned gan. *Advances in Neural Information Processing Systems*, 2021.

[36] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[37] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *ICCV*, 2021.

[38] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *CVPR*, 2021.

[39] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[40] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 2020.

[41] Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W Cohen. Re-imagen: Retrieval-augmented text-to-image generator. *arXiv:2209.14491*, 2022.

[42] 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. *IJCV*, 2024.

[43] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

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

[45] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021.

[46] Yida Chen, Fernanda Viégas, and Martin Wattenberg. Beyond surface statistics: Scene representations in a latent diffusion model. *arXiv:2306.05720*, 2023.

[47] Yizong Cheng. Mean shift, mode seeking, and clustering. *TPAMI*, 1995.

[48] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv:2011.10650*, 2020.

[49] Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. PiCIE: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *CVPR*, 2021.

[50] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? An analysis of BERT's attention. *arXiv:1906.04341*, 2019.

[51] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[52] Dorin Comaniciu and Peter Meer. Mean shift analysis and applications. In *ICCV*, 1999.

[53] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[54] Timothee Cour, Florence Benezit, and Jianbo Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR*, 2005.

[55] Victor Guilherme Turrisi da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. solo-learn: A library of self-supervised methods for visual representation learning. *Journal of Machine Learning Research*, 2022.

[56] Xili Dai, Shengbang Tong, Mingyang Li, Ziyang Wu, Michael Psenka, Kwan Ho Ryan Chan, Pengyuan Zhai, Yaodong Yu, Xiaojun Yuan, Heung-Yeung Shum, et al. Ctrl: Closed-loop transcription to an ldr via minimaxing rate reduction. *Entropy*, 2022.

[57] Paul Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *PACM-CGIT*, 1998.

[58] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[59] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021.

[60] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2015.

[61] Piotr Dollár and C. Lawrence Zitnick. Fast edge detection using structured forests. *PAMI*, 2015.

[62] Mischa Dombrowski, Hadrien Reynaud, Matthew Baugh, and Bernhard Kainz. Foreground-background separation through concept distillation from generative image foundation models. In *ICCV*, 2023.

[63] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *Advances in neural information processing systems*, 2019.

[64] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *International Conference on Learning Representations*, 2017.

[65] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *ICML*, 2021.

[66] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[67] Xiaodan Du, Nicholas Kolkin, Greg Shakhnarovich, and Anand Bhattad. Generative models: What do they know? do they know things? let's find out! *arXiv:2311.17137*, 2023.

[68] Yilun Du, Shuang Li, Yash Sharma, Josh Tenenbaum, and Igor Mordatch. Unsupervised learning of compositional energy concepts. *Advances in Neural Information Processing Systems*, 2021.

[69] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. In *International Conference on Learning Representations*, 2017.

[70] 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*, 2021.

[71] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes challenge: A retrospective. *IJCV*, 2015.

[72] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. Revisiting deep intrinsic image decompositions. In *CVPR*, 2018.

[73] Wan-Cyuan Fan, Yen-Chun Chen, DongDong Chen, Yu Cheng, Lu Yuan, and Yu-Chiang Frank Wang. Frido: Feature pyramid diffusion for complex scene image synthesis. In *AAAI*, 2023.

[74] Kirsten Fischer, David Dahmen, and Moritz Helias. Optimal signal propagation in ResNets through residual scaling. *arXiv:2305.07715*, 2023.

[75] David Forsyth and Jason J Rock. Intrinsic image decomposition using paradigms. *PAMI*, 2021.

[76] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. In *SIGGRAPH Asia*, 2017.

[77] Marc-André Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagné, and Jean-François Lalonde. Deep parametric indoor lighting estimation. In *CVPR*, 2019.

[78] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-François Lalonde. Fast spatially-varying indoor lighting estimation. In *CVPR*, 2019.

[79] Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International conference on machine learning*, pages 10929–10974. PMLR, 2023.

[80] Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics. In *NeurIPS*, 2024.

[81] A.L. Gilchrist. *Seeing Black and White*. Oxford University Press, 2006.

[82] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *PAMI*, 2016.

[83] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *ICAIS*, 2010.

[84] Melvyn A. Goodale and A. David Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 1992.

[85] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NeurIPS*, 2014.

[86] Klaus Greff, Rupesh K Srivastava, and Jürgen Schmidhuber. Highway and residual networks learn unrolled iterative estimation. In *ICLR*, 2017.

[87] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 2020.

[88] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Joshua M Susskind, and Navdeep Jaitly. Matryoshka diffusion models. In *ICLR*, 2023.

[89] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, 2022.

[90] Ronja Güldenring and Lazaros Nalpantidis. Self-supervised contrastive learning on agricultural images. *Computers and Electronics in Agriculture*, 2021.

[91] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 2017.

[92] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *ICLR*, 2022.

[93] Bharath Hariharan, Pablo Andrés Arbeláez, Ross B. Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015.

[94] Louay Hazami, Rayhane Mama, and Ragavan Thurairatnam. Efficientvdvae: Less is more. *arXiv:2203.13751*, 2022.

[95] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[96] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[97] 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*, 2020.

[98] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[99] Eric Hedlin, Gopal Sharma, Shweta Mahajan, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. Unsupervised semantic correspondence using stable diffusion. *arXiv:2305.15581*, 2023.

[100] E. Hering. *Outlines of a theory of the light sense*. 1964. Translated from the German of 1874 by L.M Hurvich and D. Jameson.

[101] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, 2007.

[102] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 2017.

[103] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Semantic object accuracy for generative text-to-image synthesis. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

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

[105] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[106] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *JMLR*, 2022.

[107] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[108] Vincent Tao Hu, David W Zhang, Yuki M Asano, Gertjan J Burghouts, and Cees GM Snoek. Self-guided diffusion models. In *CVPR*, 2023.

[109] Zhongyun Hu, Xin Huang, Yaning Li, and Qing Wang. Sa-ae for any-to-any relighting. In *ECCV*, 2020.

[110] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[111] Zhicheng Huang, Xiaojie Jin, Chengze Lu, Qibin Hou, Ming-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners. *TPAMI*, 2023.

[112] Drew A Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K Lampinen, Andrew Jaegle, James L McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. Soda: Bottleneck diffusion models for representation learning. In *CVPR*, 2024.

[113] Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *arXiv:2103.10427*, 2021.

[114] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 2015.

[115] Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. *arXiv preprint arXiv:2106.05258*, 2021.

[116] Michael Janner, Jiajun Wu, Tejas D Kulkarni, Ilker Yildirim, and Josh Tenenbaum. Self-supervised intrinsic image decomposition. In *NeurIPS*, 2017.

[117] Xu Ji, Joao F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019.

[118] Ruoxi Jiang and Rebecca Willett. Embed and emulate: Learning to estimate parameters of dynamical systems with uncertainty quantification. *Advances in Neural Information Processing Systems*, 2022.

[119] Ruoxi Jiang, Peter Y Lu, Elena Orlova, and Rebecca Willett. Training neural operators to preserve invariant measures of chaotic attractors. In *NeurIPS*, 2024.

[120] Li Jing, Jure Zbontar, and Yann LeCun. Implicit rank-minimizing autoencoder. In *NeurIPS*, 2020.

[121] Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. *arXiv:2411.02385*, 2024.

[122] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up GANs for text-to-image synthesis. In *CVPR*, 2023.

[123] Oğuzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir. 3d common corruptions and data augmentation. In *CVPR*, 2022.

[124] Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Diffusion models for zero-shot open-vocabulary segmentation. *arXiv:2306.09316*, 2023.

[125] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[126] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[127] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generati/pve adversarial networks with limited data. *Advances in neural information processing systems*, 2020.

[128] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[129] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 2021.

[130] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *TPAMI*, 2021.

[131] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

[132] Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. Rendering synthetic objects into legacy photographs. In *SIGGRAPH Asia*, 2011.

[133] Kevin Karsch, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, Hailin Jin, Rafael Fonte, Michael Sittig, and David Forsyth. Automatic scene inference for 3d object compositing. *TOG*, 2014.

[134] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *NeurIPS*, 2021.

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

[136] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[137] 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 Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[138] Peter Kocsis, Julien Philip, Kalyan Sunkavalli, Matthias Nießner, and Yannick Hold-Geoffroy. Lightit: Illumination modeling and control for diffusion models. *arXiv:2403.10615*, 2024.

[139] Peter Kocsis, Vincent Sitzmann, and Matthias Nießner. Intrinsic image diffusion for single-view material estimation. In *CVPR*, 2024.

[140] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *NeurIPS*, 2022.

[141] Iasonas Kokkinos. Pushing the boundaries of boundary detection using deep learning. *arXiv:1511.07386*, 2015.

[142] Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of BERT. In *EMNLP*, 2019.

[143] Dwight J. Kravitz, Kadharbatcha S. Saleem, Chris I. Baker, and Mortimer Mishkin. A new neural framework for visuospatial processing. *Nature Reviews Neuroscience*, 2011.

[144] Rayan Krishnan, Pranav Rajpurkar, and Eric J Topol. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, 2022.

[145] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[146] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[147] E.H. Land. Color vision and the natural image: Part i. *PNAS*, 1959.

[148] E.H. Land. Color vision and the natural image: Part ii. *PNAS*, 1959.

[149] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2017.

[150] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. FractalNet: Ultra-deep neural networks without residuals. In *ICLR*, 2017.

[151] Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. *arXiv:2305.18486*, 2023.

[152] Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *CVPR*, 2023.

[153] Bing Li, Haina Qin, Weihua Xiong, Yangxi Li, Songhe Feng, Weiming Hu, and Stephen Maybank. Ranking-based color constancy with limited training samples. *PAMI*, 2023.

[154] Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022.

[155] Jiahao Li, Greg Shakhnarovich, and Raymond A. Yeh. Adapting CLIP for phrase localization without further training. *arXiv:2204.03647*, 2022.

[156] Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. MAGE: Masked generative encoder to unify representation learning and image synthesis. In *CVPR*, 2023.

[157] Tianhong Li, Dina Katabi, and Kaiming He. Self-conditioned image generation via generating representations. *arXiv:2312.03701*, 2023.

[158] Zengyi Li, Yubei Chen, Yann LeCun, and Friedrich T Sommer. Neural manifold clustering and embedding. *arXiv preprint arXiv:2201.10000*, 2022.

[159] Zhengqi Li and Noah Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In *ECCV*, 2018.

[160] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *CVPR*, 2020.

[161] Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Meng Song, Yuhan Liu, Yu-Ying Yeh, Rui Zhu, Nitesh Gundavarapu, Jia Shi, et al. Openrooms: An open framework for photorealistic indoor scene datasets. In *CVPR*, 2021.

[162] Zhengqin Li, Jia Shi, Sai Bi, Rui Zhu, Kalyan Sunkavalli, Miloš Hašan, Zexiang Xu, Ravi Ramamoorthi, and Manmohan Chandraker. Physically-based editing of indoor scene lighting from a single image. In *ECCV*, 2022.

[163] 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 *ECCV*, 2014.

[164] Krzysztof Lis, Matthias Rottmann, Sina Honari, Pascal Fua, and Mathieu Salzmann. AttEntropy: Segmenting unknown objects in complex scenes using the spatial attention entropy of semantic segmentation transformers. *arXiv:2212.14397*, 2022.

[165] Andrew Liu, Shiry Ginosar, Tinghui Zhou, Alexei A Efros, and Noah Snavely. Learning to factorize and relight a city. In *ECCV*, 2020.

[166] Qihao Liu, Zhanpeng Zeng, Ju He, Qihang Yu, Xiaohui Shen, and Liang-Chieh Chen. Alleviating distortion in image generation via multi-resolution diffusion models. *arXiv:2406.09416*, 2024.

[167] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[168] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[169] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[170] Eric Luhman and Troy Luhman. Optimizing hierarchical image VAEs for sample quality. *arXiv:2210.10205*, 2022.

[171] Xuezhe Ma, Xiang Kong, Shanghang Zhang, and Eduard H Hovy. Decoupling global and local representations via invertible generative flows. In *International Conference on Learning Representations*, 2021.

[172] Michael Maire and Stella X Yu. Progressive multigrid eigensolvers for multiscale spectral segmentation. In *ICCV*, 2013.

[173] Michael Maire, Stella X. Yu, and Pietro Perona. Object detection and segmentation from joint embedding of parts and pixels. In *ICCV*, 2011.

[174] Michael Maire, Takuya Narihira, and Stella X. Yu. Affinity CNN: Learning pixel-centric pairwise relations for figure/ground embedding. In *CVPR*, 2016.

[175] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[176] Marina Meila and Jianbo Shi. Learning segmentation by random walks. In *NeurIPS*, 2000.

[177] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *CVPR*, 2022.

[178] Mortimer Mishkin, Leslie G. Ungerleider, and Kathleen A. Macko. Object vision and spatial vision: two cortical pathways. *Trends in Neurosciences*, 1983.

[179] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[180] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: An overview. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, 2019.

[181] Lukas Murmann, Michael Gharbi, Miika Aittala, and Fredo Durand. A dataset of multi-illumination images in the wild. In *CVPR*, 2019.

[182] Thomas Nestmeyer and Peter V Gehler. Reflectance adaptive filtering improves intrinsic image estimation. In *CVPR*, 2017.

[183] Thomas Nestmeyer, Jean-François Lalonde, Iain Matthews, Epic Games, Andreas Lehrmann, and AI Borealis. Learning physics-guided face relighting under directional light. In *CVPR*, 2020.

[184] Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. In *ICLR*, 2019.

[185] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv:2112.10741*, 2021.

[186] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[187] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *JMLR*, 2021.

[188] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[189] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[190] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[191] Adeel Pervez and Efstratios Gavves. Variance reduction in hierarchical variational autoencoders. 2020.

[192] Julien Philip, Michaël Gharbi, Tinghui Zhou, Alexei A Efros, and George Drettakis. Multiview relighting using a geometry-aware network. *TOG*, 2019.

[193] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[194] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[195] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[196] Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism of feature learning in deep fully connected networks and kernel machines that recursively learn features. *arXiv:2212.13881*, 2022.

[197] Adityanarayanan Radhakrishnan, Mikhail Belkin, and Dmitriy Drusvyatskiy. Linear recursive feature machines provably recover low-rank matrices. *arXiv:2401.04553*, 2024.

[198] Ravi Ramamoorthi and Pat Hanrahan. On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *JOSAA*, 2001.

[199] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.

[200] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv:2204.06125*, 2022.

[201] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[202] Xiaofeng Ren and Liefeng Bo. Discriminatively trained sparse code gradients for contour detection. In *NeurIPS*, 2012.

[203] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021.

[204] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[205] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[206] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 1987.

[207] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *ESPC*, 2007.

[208] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

[209] Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M Weber. Cads: Unleashing the diversity of diffusion models through condition-annealed sampling. *arXiv:2310.17347*, 2023.

[210] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

[211] Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving gans using optimal transport. *arXiv preprint arXiv:1803.05573*, 2018.

[212] Ayush Sarkar, Hanlin Mai, Amitabh Mahapatra, Svetlana Lazebnik, David A Forsyth, and Anand Bhattad. Shadows don't lie and lines can't bend! generative models don't know projective geometry... for now. In *CVPR*, 2024.

[213] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 Conference Proceedings*, 2022.

[214] Pedro Savarese and Daniel Figueiredo. Residual gates: A simple mechanism for improved network optimization. In *ICLR*, 2017.

[215] Gerald E. Schneider. Two visual systems: Brain mechanisms for localization and discrimination are dissociated by tectal and cortical lesions. *Science*, 1969.

[216] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022.

[217] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[218] Soumyadip Sengupta, Brian Curless, Ira Kemelmacher-Shlizerman, and Steven M Seitz. A light stage on every desk. In *ICCV*, 2021.

[219] Hyun Seok Seong, WonJun Moon, SuBeen Lee, and Jae-Pil Heo. Leveraging hidden positives for unsupervised semantic segmentation. In *CVPR*, 2023.

[220] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *PAMI*, 2000.

[221] Zhan Shi, Xu Zhou, Xipeng Qiu, and Xiaodan Zhu. Improving image captioning with better use of captions. In *ACL*, 2020.

[222] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[223] Mannat Singh, Quentin Duval, Kalyan Vasudev Alwala, Haoqi Fan, Vaibhav Aggarwal, Aaron Adcock, Armand Joulin, Piotr Dollár, Christoph Feichtenhofer, Ross Girshick, et al. The effectiveness of MAE pre-pretraining for billion-scale pretraining. *arXiv:2303.13496*, 2023.

[224] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

[225] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *NeurIPS*, 2016.

[226] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, 2020.

[227] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *NeurIPS*, 2020.

[228] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[229] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023.

[230] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv:1412.6806*, 2014.

[231] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv:1505.00387*, 2015.

[232] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *NeurIPS*, 2015.

[233] Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *TOG*, 2019.

[234] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv:1312.6199*, 2013.

[235] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[236] Yuhta Takida, Yukara Ikemiya, Takashi Shibuya, Kazuki Shimada, Woosung Choi, Chieh-Hsin Lai, Naoki Murata, Toshimitsu Uesaka, Kengo Uchida, Wei-Hsiang Liao, et al. Hq-vae: Hierarchical discrete representation learning with variational bayes. *arXiv:2401.00365*, 2023.

[237] Akinori Tanaka. Discriminator optimal transport. *Advances in Neural Information Processing Systems*, 2019.

[238] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *NeurIPS*, 2023.

[239] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised CNN segmentation. In *CVPR*, 2018.

[240] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: A family of highly capable multimodal models. *arXiv:2312.11805*, 2023.

[241] Changyao Tian, Chenxin Tao, Jifeng Dai, Hao Li, Ziheng Li, Lewei Lu, Xiaogang Wang, Hongsheng Li, Gao Huang, and Xizhou Zhu. ADDP: Learning general representations for image recognition and generation with alternating denoising diffusion process. In *ICLR*, 2024.

[242] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv:2404.02905*, 2024.

[243] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 2020.

[244] Colwyn B. Trevarthen. Two mechanisms of vision in primates. *Psychologische Forschung*, 1968.

[245] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. In *NeurIPS*, 2020.

[246] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017.

[247] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008.

[248] 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*, 2017.

[249] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[250] H. von Helmholtz. *Helmholtz' treatise on physiological optics*. 1924-1925. Translated from the 3rd German Edition of 1867, edited by J.P Southall.

[251] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

[252] Feng Wang, Tao Kong, Rufeng Zhang, Huaping Liu, and Hang Li. Self-supervised learning by estimating twin class distributions. *arXiv preprint arXiv:2110.07402*, 2021.

[253] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *CVPR*, 2023.

[254] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M. Alvarez. FreeSOLO: Learning to segment objects without annotations. In *CVPR*, 2022.

[255] Xudong Wang, Rohit Girdhar, Stella X. Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *CVPR*, 2023.

[256] Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation. In *CVPR*, 2024.

[257] Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L. Crowley, and Dominique Vaufreydaz. TokenCut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *arXiv:2209.00383*, 2022.

[258] Yufei Wang, Renjie Wan, Wenhan Yang, Haoliang Li, Lap-Pui Chau, and Alex Kot. Low-light image enhancement with normalizing flow. In *AAAI*, 2022.

[259] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.

[260] Zian Wang, Jonah Philion, Sanja Fidler, and Jan Kautz. Learning indoor inverse rendering with 3d spatially-varying lighting. In *ICCV*, 2021.

[261] Henrique Weber, Mathieu Garon, and Jean-François Lalonde. Editable indoor lighting estimation. In *ECCV*, 2022.

[262] Yibing Wei, Abhinav Gupta, and Pedro Morgado. Towards latent masked image modeling for self-supervised visual representation learning. *arXiv preprint arXiv:2407.15837*, 2024.

[263] Christoph Witzel and Karl R. Gegenfurtner. Color perception: Objects, constancy, and categories. *Annu Rev Vis Sci*, 2018.

[264] Yi-Lun Wu, Hong-Han Shuai, Zhi-Rui Tam, and Hong-Yu Chiu. Gradient normalization for generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[265] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision*, 2018.

[266] 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*, 2018.

[267] Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. *arXiv preprint arXiv:2303.09769*, 2023.

[268] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. GroupViT: Semantic segmentation emerges from text supervision. In *CVPR*, 2022.

[269] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, 2023.

[270] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[271] Hao-Hsiang Yang, Wei-Ting Chen, and Sy-Yen Kuo. S3net: A single stream structure for depth guided image relighting. In *CVPR*, 2021.

[272] Xingyi Yang and Xinchao Wang. Diffusion model as representation learner. In *ICCV*, 2023.

[273] Kun Yi, Yixiao Ge, Xiaotong Li, Shusheng Yang, Dian Li, Jianping Wu, Ying Shan, and Xiaohu Qie. Masked image modeling with denoising contrast. In *ICLR*, 2023.

[274] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv:2206.10789*, 2022.

[275] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multimodal models: Pretraining and instruction tuning. *arXiv:2309.02591*, 2023.

[276] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv:2410.06940*, 2024.

[277] Stella X. Yu and Jianbo Shi. Segmentation given partial grouping constraints. *PAMI*, 2004.

[278] Stella X. Yu, Ralph Gross, and Jianbo Shi. Concurrent object recognition and segmentation by graph partitioning. In *NeurIPS*, 2002.

[279] Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. *NeurIPS*, 2020.

[280] Ye Yu and William AP Smith. Inverserendernet: Learning single image inverse rendering. In *CVPR*, 2019.

[281] Ye Yu, Abhimitra Meka, Mohamed Elgharib, Hans-Peter Seidel, Christian Theobalt, and William AP Smith. Self-supervised outdoor scene relighting. In *ECCV*, 2020.

[282] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, 2021.

[283] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021.

[284] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[285] Mingtian Zhang, Tim Z Xiao, Brooks Paige, and David Barber. Improving vae-based representation learning. *arXiv preprint arXiv:2205.14539*, 2022.

[286] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Proceedings of the European conference on computer vision*, 2016.

[287] Xiao Zhang and Michael Maire. Self-supervised visual representation learning from hierarchical grouping. *Advances in Neural Information Processing Systems*, 2020.

[288] Xiao Zhang and Michael Maire. Structural adversarial objectives for self-supervised representation learning. *arXiv:2310.00357*, 2023.

[289] Xiao Zhang, William Gao, Seemandhar Jain, Michael Maire, David Forsyth, and Anand Bhattad. Latent intrinsics emerge from training to relight. *Advances in Neural Information Processing Systems*, 37:96775–96796, 2024.

[290] Xiao Zhang, Ruoxi Jiang, William Gao, Rebecca Willett, and Michael Maire. Residual connections harm generative representation learning. *arXiv preprint arXiv:2404.10947*, 2024.

[291] Xiao Zhang, David Yunis, and Michael Maire. Deciphering 'what' and 'where' visual pathways from spectral clustering of layer-distributed neural representations. In *CVPR*, 2024.

[292] Xiao Zhang, Ruoxi Jiang, Rebecca Willett, and Michael Maire. Nested diffusion models using hierarchical latent priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2502–2512, 2025.

[293] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from generative models. *arXiv:1702.08396*, 2017.

[294] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from CLIP. In *ECCV*, 2022.

[295] Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W Jacobs. Deep single-image portrait relighting. In *ICCV*, 2019.

[296] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT pre-training with online tokenizer. In *ICLR*, 2022.

[297] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[298] Ligeng Zhu, Ruizhi Deng, Michael Maire, Zhiwei Deng, Greg Mori, and Ping Tan. Sparsely aggregated convolutional networks. In *ECCV*, 2018.

[299] Adrian Ziegler and Yuki M. Asano. Self-supervised learning of object parts for semantic segmentation. In *CVPR*, 2022.

APPENDIX A

RESIDUAL CONNECTIONS HARM GENERATIVE REPRESENTATION LEARNING

A.1 Training and Evaluation Details

A.1.1 Model training.

Our training configurations primarily followed the guidelines established by He et al. [98]. In the ImageNet-1K experiment, our model was trained for 800 epochs, utilizing the AdamW [169] optimizer with a constant weight decay of 5e-2 for a batch size of 1024. We set the maximum learning rate to 6e-4. Initially, the learning rate started at 0 and linearly increased to its maximum over the first 40 epochs, after which it followed a cosine schedule to gradually decrease to zero by the end of the training period. It is worth noting that the learning rate per sample, or effective learning rate, in our setup matched that of He et al. [98], although our maximum learning rate was set lower due to our batch size being a quarter of theirs. We applied random resizing, cropping, and horizontal flipping during training as part of our augmentation scheme. To enhance the quality of the learned representations in most experiments, we employed the normalized pixel loss, as proposed by [98]. In the ImageNet-100 experiment, we employed the identical training configuration used in the ImageNet-1K experiments. We train our model with 4 NVIDIA A40 GPUs and a completed trianing usually takes 20 hours on ImageNet-100 and 200 hours on ImageNet-1k.

A.1.2 Evaluation with Linear Probing.

For the ImageNet-1k dataset, we use the exact same evaluation protocols employed in He et al. [98], which includes random data augmentation.

For the ImageNet-100 dataset, we employed a simpler evaluation protocol: We train the linear classifier with a batch size of 1024 for 200 epochs, where the learning rate starts at 1e-2 and then decays towards 0 using a cosine scheduler. During this evaluation, we do not apply any data augmentation.

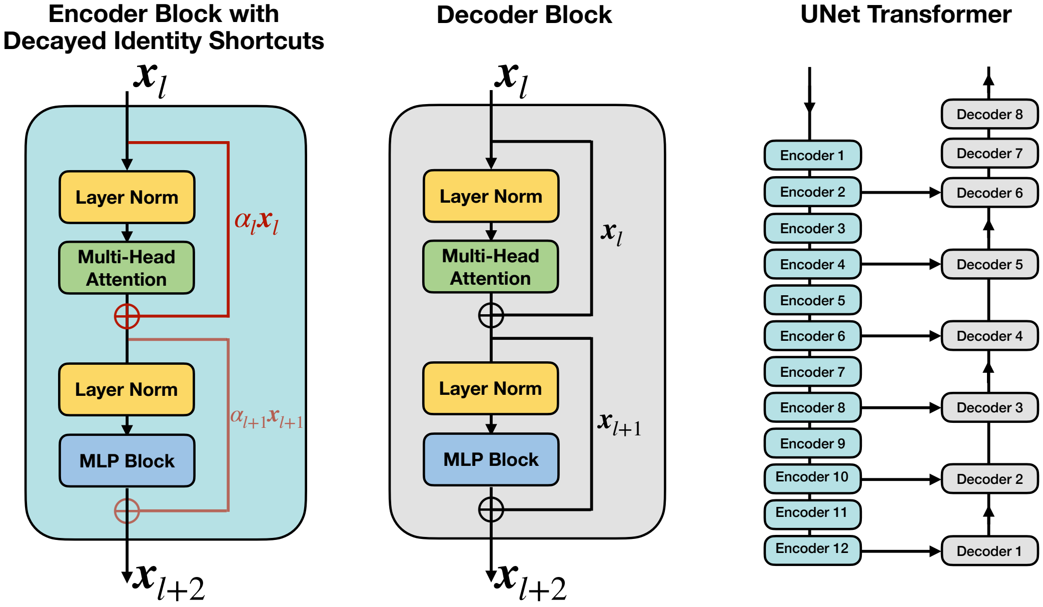


Figure A.1: We present our enhanced UNet Transformer architecture for Masked Auto-encoder. (1) ***Left***: Our customized encoder blocks, equipped with our proposed decay identity shortcuts. (2) ***Middle***: Standard transformer blocks as the decoder blocks. (3) ***Right***: We incorporate the decay identity shortcuts exclusively within the encoder blocks of our UNet transformer and employ standard transformer blocks for the decoder. To support abstract representation learning at the bottleneck, *i.e.,* the last layer of the Encoder 12, we adopt the UNet [205] architecture and create skip connections that transmit every other encoder feature directly to the decoder.

A.1.3 Modified Architecture

We present a visualization of our UNet transformer design, as outlined in Section 2.3.2, in Fig. A.1. It's important to note that decayed identity shortcuts are exclusively implemented within the encoder block. Additionally, we establish skip connections from alternating blocks in the encoder to the decoder, following the UNet [205] architecture's design principles.

Layer Index	1	2	3	4	5	6	7	8	9	10	11	12
Attention	0.993	0.947	0.982	0.766	0.992	0.795	0.988	0.849	0.998	0.723	0.811	0.488
FFN	0.989	0.926	0.961	0.620	0.961	0.442	0.711	0.322	0.810	0.475	0.637	0.353

Table A.1: Learnable α_l values for different model layers. In contrast to our proposed linear scheduler, learnable α_l does not exhibit a consistent decay pattern across network depth.

A.1.4 Learnable α_l over network layers

In the ablation study of learnable α_l, we apply no additional regularization beyond standard weight decay to the model parameters. Table A.1 presents the α_l values for each layer. The results do not reveal any meaningful pattern across network depth.

A.2 Further experiments

A.2.1 Reconstruction quality.

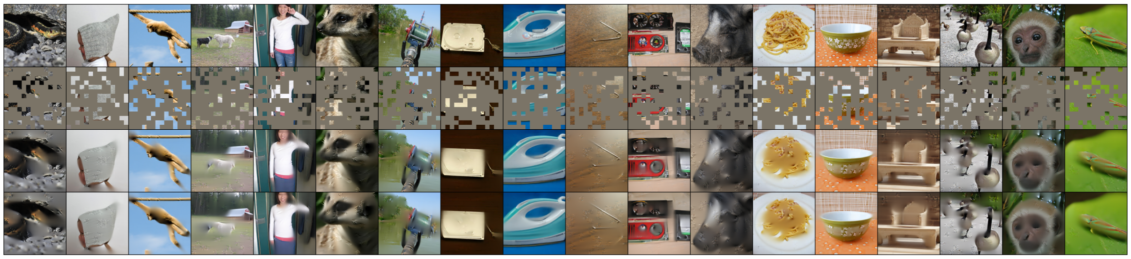
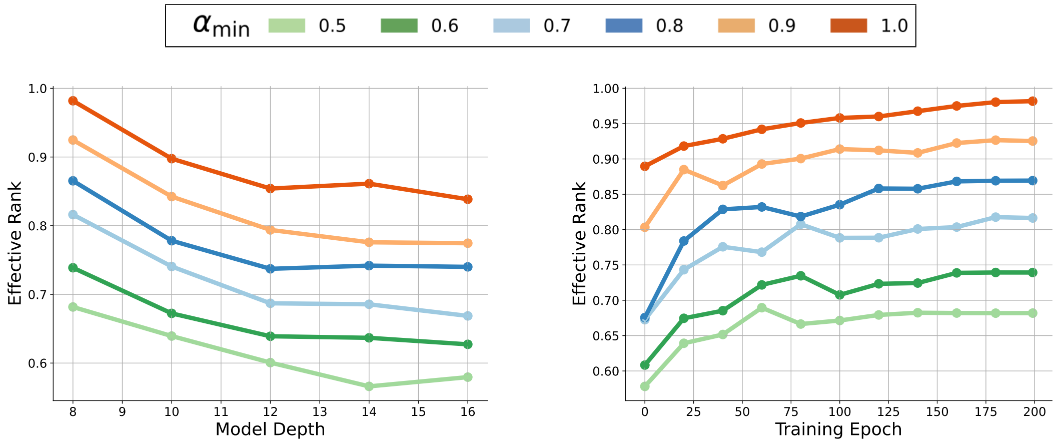


Figure A.2: **Qualitative comparison of images reconstructed by MAE with and without our method.** We observe our method learns features with higher linear probing accuracy without compromising reconstruction quality. Row 1: ground truth test image. Row 2: images masked at 75%. Row 3: reconstructions with our method. Row 4: reconstructions with baseline MAE.

We qualitatively evaluate test images reconstructed by an MAE using our framework and images reconstructed by the original MAE. We show the reconstructed images in Figure A.2. While the focus of our work is entirely to improve the representations learned by an encoder, we observe that our framework does not harm the reconstructions. Hence, there is no qualitative tradeoff for our increase in linear probing accuracy.

A.2.2 Abstraction and Low-rank in the Supervised Setting



(a) Effective rank of ResNet for different depths at the convergence of the training.

(b) Effective rank of ResNet over training epoch.

Figure A.3: **Dynamics of the feature rank in the supervised setup.** We train ResNet models for a supervised classification task on a small subset of ImageNet. And visualize (a) effective rank across different depths at convergence and (b) training dynamics of effective rank over time for various α_{min}. In (a) we see that at convergence, our method consistently decreases the feature rank with various depth and, in (b), this pattern is also shown for standard ResNet model at every stage of training.

In this experiment, we modify the standard ResNet-18 model to experiment with different depth models. By default, the ResNet-18 has a total of 8 residual blocks that are equally distributed into 4 layers. To increase model depth, we repeat residual blocks in the 3rd layer to obtain models varying between 8 and 16 total layers. At convergence, we observe that the models of different depths achieve a similar test accuracy. However, despite similar accuracies, in Figure A.3a, which visualizes the effective rank over depth for different values of α_{min}, we see that the effective rank decreases over depth. Furthermore, smaller values of α_{min} consistently lead to features with lower effective rank.

Next, in Figure A.3b, we try to verify our conjecture by visualizing the evolution of effective rank during training when choosing different α_{min} in our method. For this experiment, we choose to train the standard ResNet-18 using our decayed identity shortcuts. In this setup, we observe that the optimal choice of α_{min} slightly improves the test accuracy of the classification network:

94.4% with $\alpha_{\min} = 0.7$ *vs.* 93.6% with $\alpha_{\min} = 1.0$. We observe that the effective rank of the final features decreases with decreasing $\alpha_{\min}$. This supports our hypothesis that (1) decayed identity shortcuts substantially decrease the rank of bottleneck features and (2) decreasing feature rank may help improve learned features.

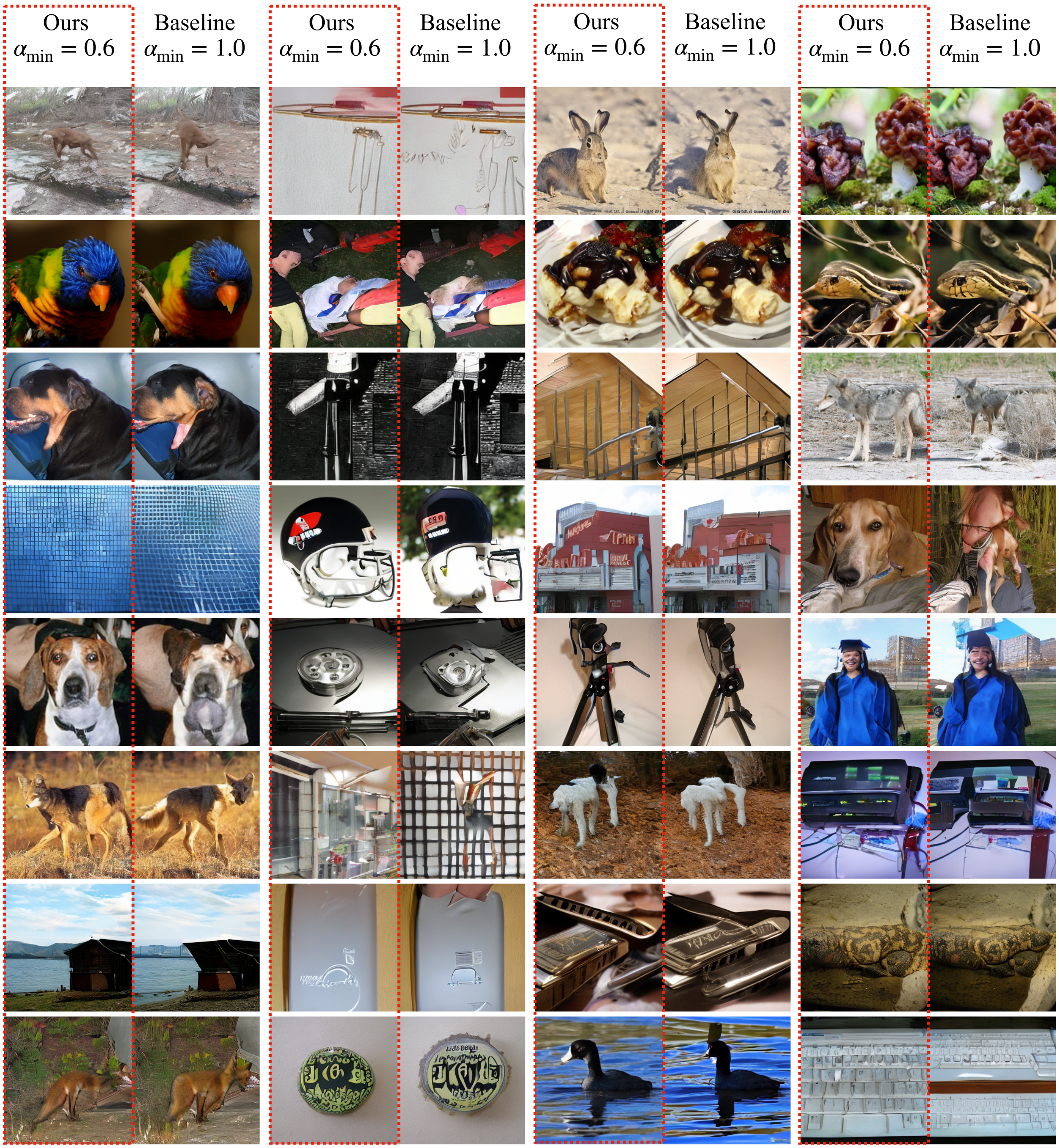

Ours
$\alpha_{\text{min}} = 0.6$
Baseline
$\alpha_{\text{min}} = 1.0$
Ours
$\alpha_{\text{min}} = 0.6$
Baseline
$\alpha_{\text{min}} = 1.0$
Ours
$\alpha_{\text{min}} = 0.6$
Baseline
$\alpha_{\text{min}} = 1.0$
Ours
$\alpha_{\text{min}} = 0.6$
Baseline
$\alpha_{\text{min}} = 1.0$

Figure A.4: **Qualitative comparison of images generated by diffusion models.** Our method, decayed identity shortcuts with $\alpha_{\text{min}} = 0.6$, shows improved representation learning and produces higher-quality generated images compared to the baseline, which employs full residual connections ($\alpha_{\text{min}} = 1.0$).

APPENDIX B

STRUCTURAL ADVERSARIAL OBJECTIVES FOR SELF-SUPERVISED REPRESENTATION LEARNING

B.1 Details of Dataset and Model

Datasets. We focus on three benchmark datasets: CIFAR-10, CIFAR-100 [145] and ImageNet-10.

ImageNet-10: We follow Chang et al. [36] to select 10 categories from the ImageNet dataset [58], resulting in 13,000 training images and 500 validation images. During training, we only perform spatial augmentation, including random spatial cropping and horizontal flipping, followed by resizing images to 128x128 resolution to match the generated images. During testing, we resize the images to align the smaller edge to 144 pixels, followed by central cropping to produce a 128x128 output.

CIFAR-10/100: During training, we apply the same augmentation strategy as in ImageNet-10 but produce 32x32 images. During testing, we do not perform cropping.

For compared methods, we keep their default augmentation strategy. On ImageNet-10, we resized their augmented images to 128x128. For all methods, we learn in an unsupervised manner on the training split and evaluate on the validation split.

In CIFAR-10/100 experiments, we use default configurations from SOLO-Learns [55], an open source library providing heavily tuned configurations for multiple state-of-the-art self-supervised methods. In ImageNet-10 experiments, we train competing approaches using the suggested hyperparameters for ImageNet-100, but extend the total epochs to 1000 for sufficient convergence. For fair comparison, we run these methods with our modified backbone and resize input images to 128x128.

Model details: discriminator. We construct our discriminator using ResNet-18 [95] and perform several modifications to make it cooperate reasonably with the generator. Inspired by the discriminator configuration in BigGAN [28], we perform spatial reduction only within the residual block and replace all stride two convolution layers with average pooling followed by stride one convolution. We remove the first max-pooling layer and switch the first convolution layer to a 3x3 kernel with a 1x1 stride to keep the resolution unchanged before the residual block.

To maintain a substantial downsample rate in ImageNet-10 images, we duplicate the first residual block and enable a spatial reduction in all blocks to reach a 32x downsampling. On CIFAR-10/100, we preserve the default setting for residual blocks. As our proposed smoothness term regularizes each sample, we replace all BatchNorm layers [114] with GroupNorm [265], specifying 16 channels as a single group; this prevents batch-wise interaction. We also remove the first normalization layer in each block, as doing so produces better results. We replace ReLU with ELU [51] activations for broader non-linear support on negative values.

Model details: generator. We adapt the generator configuration from BigGAN-deep[28]. Specifically, we take their model for 32x32 images on CIFAR, and additionally increase the base channels to 128 to prevent image generation from being the system bottleneck. For ImageNet-10, we replicate their settings for 128x128 images.

B.2 Compared Self-Supervised Learning Methods

We evaluate the representations produced by our method in comparison to those produced by the following state-of-the-art self-supervised learning methods:

- SimCLR [40] optimizes the InfoNCE loss, maximizing feature similarity across views while repulsing all the images.
- NNCLR [70] samples nearest neighbors from the data set using cross-view features and treats them as positives for InfoNCE objectives. We additionally run a baseline, denoted NNCLR (same views) in Figure 3.1, by removing the augmented view and directly maximizing the similarity between image features and their nearest neighbor.

- SWAV [33] maximizes view consistent objectives using clustering-based targets; it balances the categorical assignment using sinkhorn iterations.

- DINO [34] optimizes clustering-based across-views objectives via knowledge distillation and proposes sharpening and centering techniques to prevent collapsing.

- BYOL [87] only contains the maximizing term and adopts a momentum-updated Siamese model to process augmented input to prevent collapsed solutions.

In MAE [98], we employ a VIT-small model, training it with default masking ratio and a patch-size of 4 for CIFAR experiments and 8 for ImageNet-10 experiments.

In DDPM [105], we use unconditional model and train it with default hyper-parameters. Feature are extracted from the second decoder block with noise level at t = 11, following the optimal configurations of Xiang et al. [267].

B.3 Qualitative Comparison

We provide visualization of generated images for the following configurations:

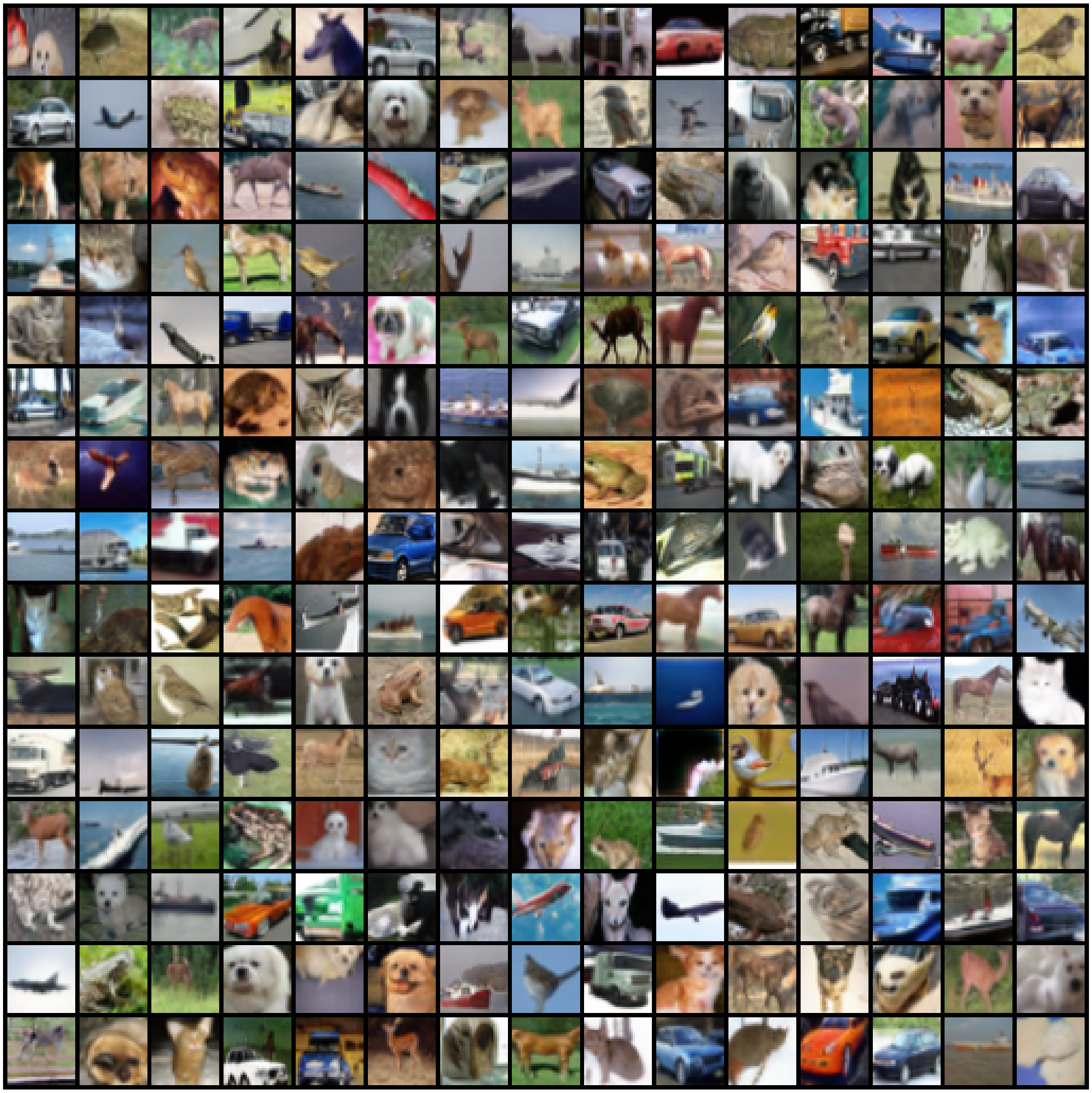


Figure B.1: Randomly generated images from GAN trained with our full objectives.

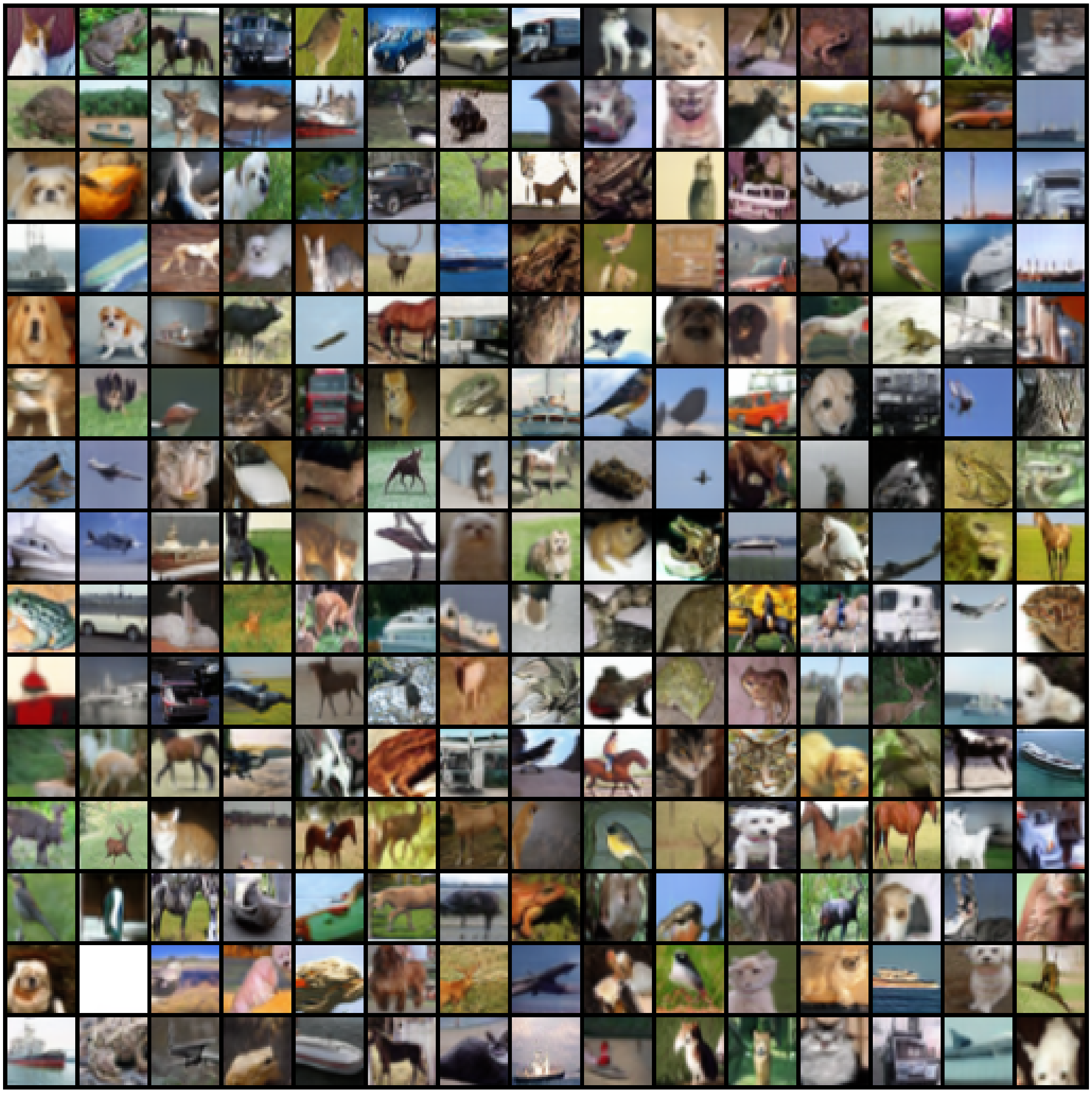


Figure B.2: Randomly generated images from GAN trained with Eqn. 3.2 only, JSD.

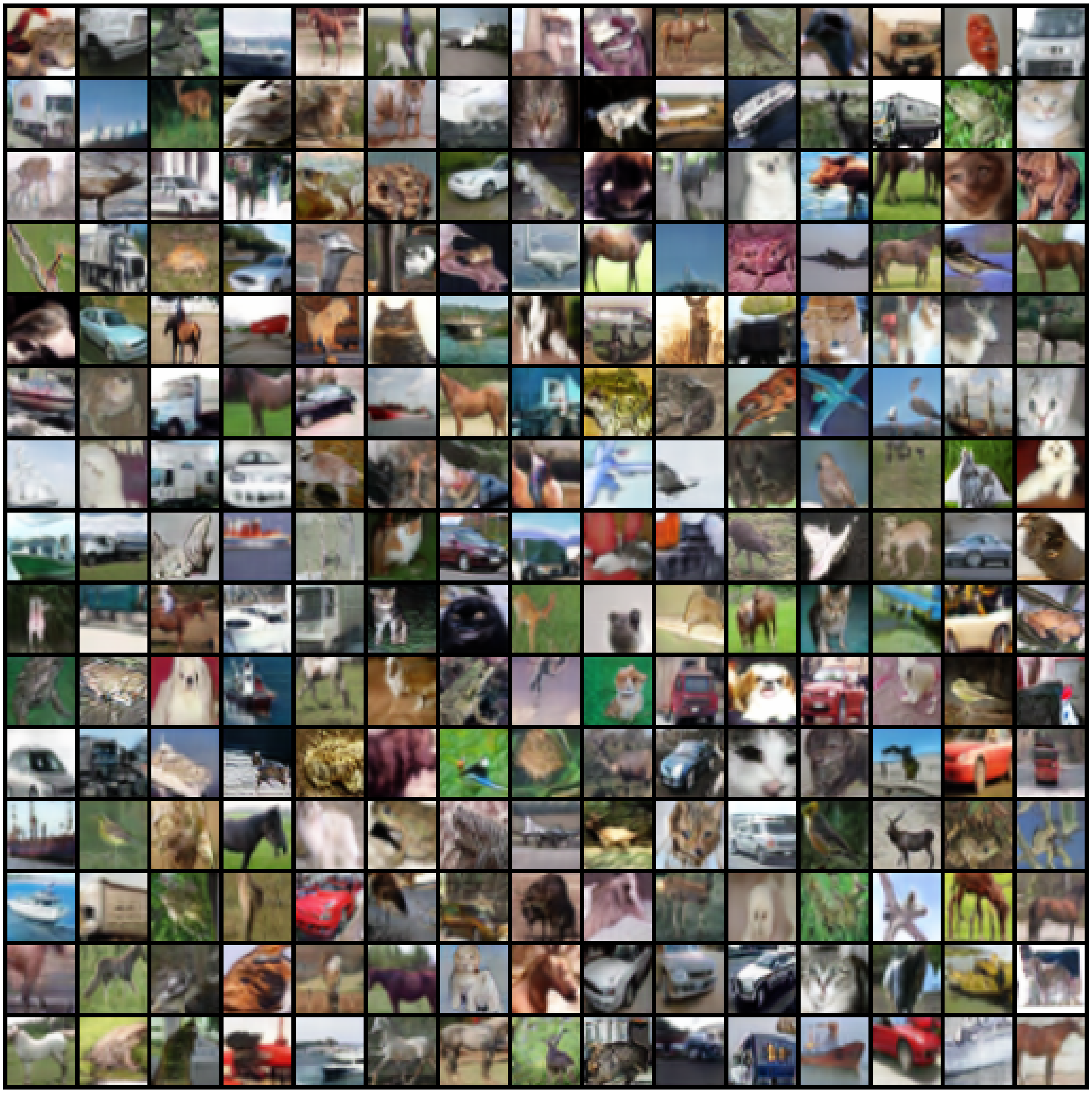


Figure B.3: Randomly generated images from GAN trained with Hinge Loss.

- Figure B.1: Results of training with our full objectives (our method):

$$\mathcal{L}^{\text{Full}} := \mathcal{L}_{\text{Gaussian}} + \lambda_c \mathcal{L}_{\text{cluster}} + \lambda_s \mathcal{L}_{\text{reg}}.$$

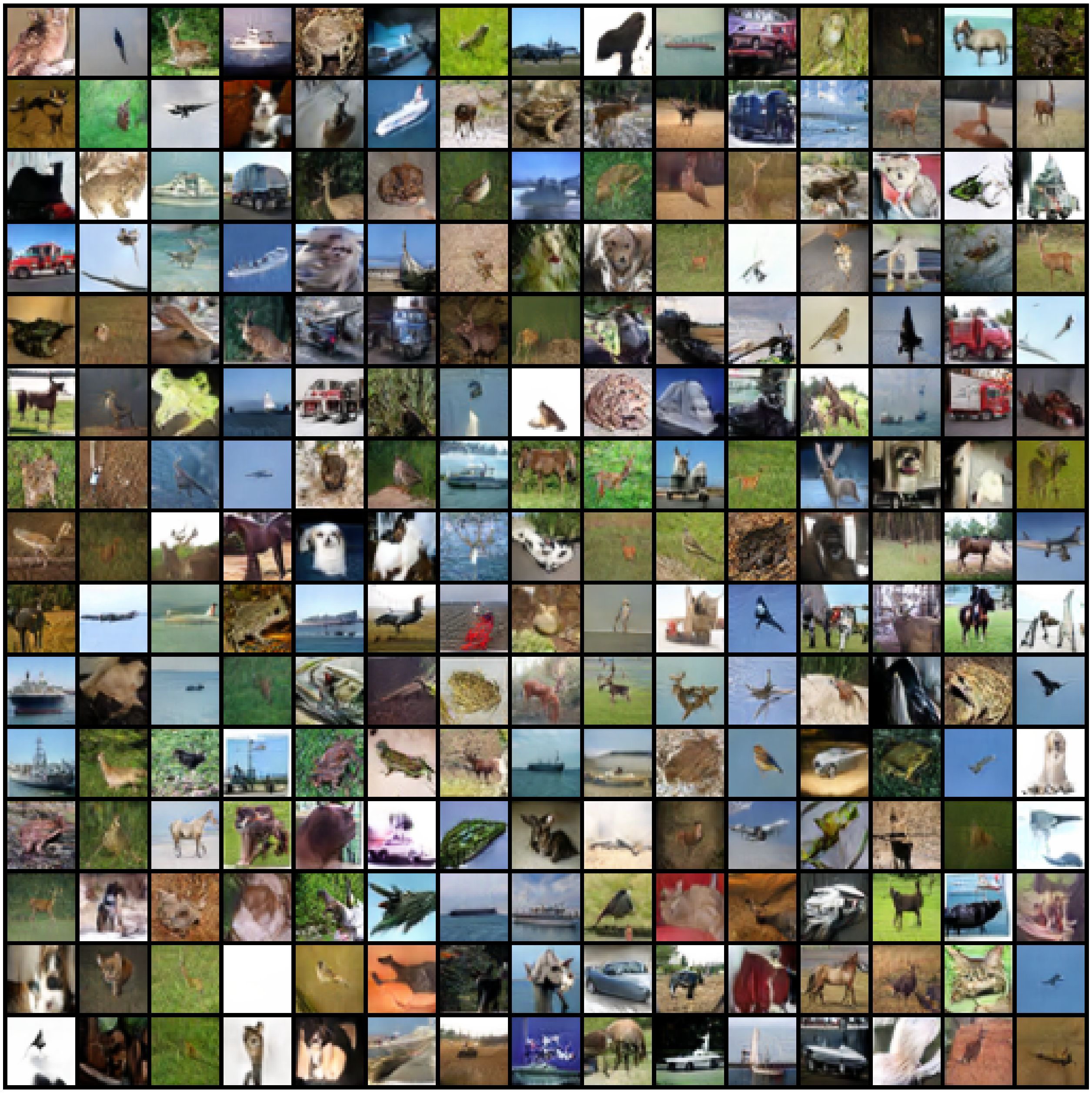


Figure B.4: Randomly generated images from unconditional BigGAN [28].

- Figure B.2: Results of training with Equation 3.2 only, JSD:

$$\mathcal{L}^{\text{JSD}} := \mathcal{L}_{\text{Gaussian}} + \lambda_s \mathcal{L}_{\text{reg}}.$$

- Figure B.3: Results of training with Hinge Loss.

  To train with Hinge loss, we change discriminator to output a scalar: $\boldsymbol{D}(\boldsymbol{x}) \in \mathbb{R}$ and optimize the hinge loss defined as follows:

$$\begin{aligned}
\mathcal{L}^{\text{Hinge}} &:= \mathcal{L}_{\boldsymbol{D}}^{\text{Hinge}} + \mathcal{L}_{\boldsymbol{G}}^{\text{Hinge}} + \lambda_s \mathcal{L}_{\text{reg}}, \\
\mathcal{L}_{\boldsymbol{D}}^{\text{Hinge}} &:= \max_{\boldsymbol{D}} \left(\min\left(0, -1 + \boldsymbol{D}\left(\boldsymbol{x}\right)\right) - \min\left(0, -1 - \boldsymbol{D}\left(\hat{\boldsymbol{x}}\right)\right)\right), \\
\mathcal{L}_{\boldsymbol{G}}^{\text{Hinge}} &:= \min_{\boldsymbol{G}} -\boldsymbol{D}(\hat{\boldsymbol{x}}).
\end{aligned}$$

- Figure B.4: Results of BigGAN [28].

Conclusion. We observe that training with full objectives (our method) achieves the best quality and diversity in generated images.

APPENDIX C

DECIPHERING ‘WHAT’ AND ‘WHERE’ VISUAL PATHWAYS FROM SPECTRAL CLUSTERING OF LAYER-DISTRIBUTED NEURAL REPRESENTATIONS

C.1 Additional Qualitative Results

We provide additional visualizations of the extracted eigenvectors for both the COCO and Cityscapes datasets in Figures C.1 and C.2. These visualizations follow the same methodology as in Figures 4.5 and 4.6, where eigenvectors $\boldsymbol{U}$ are rendered in descending order in groups of three. Each channel of the RGB image corresponds to the value of a particular eigenvector at that coordinate.

C.2 Per-Image Experimental Details

C.2.1 Data Preprocessing

For models besides Stable Diffusion [204], and Masked Autoencoder (MAE) [98] image inputs are resized to have a short dimension of 448 pixels. This requires change to the resolution of the learned positional embeddings, which we do through bicubic interpolation, similar to MaskCLIP [294]. In the case of Stable Diffusion, we instead resize images to 512x512 to match the input dimensions of the original model. For MAE, we resize images to 224x224, then upsample the internal query and key matrices to match the spatial resolution of CLIP and DINO.

C.2.2 Optimization

We optimize features with Adam [135], with a learning rate of 3e-4 or 1e-3, and default PyTorch [188] betas $(0.9, 0.999)$. We take a number of gradient steps to convergence that depends on the model (1000 for CLIP, 2000 for others), but we typically find that 1000 steps is sufficient.

Model	Affinity Source	Mask	mIoU
Stable Diff. 1.4 [204]	All Attentions	Ours + K-Means	**0.82**
CLIP ViT-B/16 [195]	All Attentions	Ours + K-Means	**0.78**
CLIP ViT-B/16 [195]	Final Features	K-Means	0.57
CLIP ViT-B/16 [195]	Final Features	Ncut + K-Means [257]	0.45
DINO ViT-S/16 [34]	All Attentions	Ours + K-Means	**0.78**
DINO ViT-S/16 [34]	Final Attentions	Ncut + K-Means [257]	0.58
DINO ViT-S/16 [34]	Final Features	K-Means	0.74
DINO ViT-S/16 [34]	Final Features	Ncut + K-Means [257]	0.73
DINO ViT-S/16 [34]	Final Features	MaskCut[255]	0.64
MAE ViT-B/16 [98]	All Attention	Ours + K-Means	**0.74**
MAE ViT-B/16 [98]	Final Features	Ncut + K-Means [34]	0.62
MAE ViT-B/16 [98]	Final Features	K-Means	0.48

Table C.1: **Oracle decoding on PASCAL VOC [71]** . Compared with several strong baselines [257, 255] applied to single-level features, our method can consistently extract accurate segmentation. Our method works well even for models like CLIP [195] and MAE [98], whose final layer features are not discriminative enough for segmentation. Our method is agnostic to the location of information, so we avoid this difficulty.

Timing information for our method is available in Table C.2.

Unlike other models, where there is only a single set of attention matrices per image, the sampling of t in the forward pass of Stable Diffusion introduces more noise and significantly more computation into the optimization. To address this, we cache attention matrices in a buffer of 5 at a time, where the chance to sample a new set of attention matrices is 1/4, and the oldest set in the buffer is replaced by this sample. We also accumulate gradients for 20 backward passes before taking an optimizer step.

C.2.3 Baselines

To extract regions from TokenCut [257] and MaskCut [255], a single affinity matrix is required. One choice is an affinity matrix constructed from features of the final layer, which is the original proposed matrix for these methods. Another is the final layer's attention matrix. A third alternative is to compute an average over all attention matrices across layers, so as to better compare to our method. We found the third option often led to an ill-conditioned matrix, which could not be solved.

Model	**Runtime (seconds)**
Stable Diffusion 1.4 (w/ buffer) [204]	67
Stable Diffusion 1.4 (w/o buffer) [204]	155
DINO ViT-S/16 [34]	40
MAE/CLIP ViT-B/16 [195]	54

Table C.2: **Computation time across models.** We benchmark region computation time for 1000 optimization steps using different models on an NVIDIA A40. 1000 steps are often not required for good results, thus it may be possible to significantly accelerate the pipeline.

Consequently, we present results for the first two choices. For methods except TokenCut, we find best results with $m = 15$ eigenvectors. For TokenCut we found the performance with $m = 15$ to be subpar, so we use $m = 8$ instead. Quantitative results are available in Table C.1. Qualitative results comparing decoding methods can be seen in Figure C.3. See Figure 4.3 for comparison between regions extracted from different models.

C.2.4 Computational Cost

Table C.2 shows the computational cost of running our method, benchmarked on an NVIDIA A40. The extremely long computation time for Stable Diffusion is due to many evaluations of the model during optimization, instead of simply caching the attention matrices from a single forward pass.

C.3 Full-Dataset Experimental Details

C.3.1 Data Preprocessing

All experiments take place on COCO-Stuff [163, 29] and Cityscapes [53]. We follow the same preprocessing protocol as adopted in PiCIE [49] and STEGO [92]: images are first resized so the minor edge is 320px and then cropped in the center to produce square images.

C.3.2 Optimization

In the per-image setting we choose each head to be an independent affinity graph, but that leads to extremely expensive experiments at the full-dataset level. To control this expense, we experiment with a few alternatives: considering each head independently and sampling random layers and heads per iteration of optimization, or concatenating the features for each head into one large vector, which reduces the number of graphs by a factor of 8. The second ultimately led to better results. Due to prohibitive memory costs, we also only consider attention layers with resolutions of 32x32 or coarser. This avoids the large graphs constructed by layers with 64x64 resolution. Due to the prohibitive cost associated with optimizing one set of features per image in the dataset, we restrict our dataset-level clustering to the validation set only.

C.3.3 Evaluation

Unsupervised semantic segmentation. We consider $\boldsymbol{X}_{\text{ortho}}$ as features for our method. We also compare with several baseline methods by collecting backbone features from a number of different models: STEGO, DINO and Stable Diffusion. For Stable Diffusion we choose the most semantic features in the model, as measured by semantic correspondence performance in prior work [238]. For DINO we take features at the last layer, like prior work [257, 255, 92]. For STEGO, we use output just before the linear head that projects to the number of clusters.

After obtaining features, we cluster with $K = 27$, the number of ground truth categories in both datasets, for K-Means over $\boldsymbol{X}_{\text{ortho}}$. We report results with mIoU and compare to other methods in Table 4.2.

X-Y coordinate regression. After extracting features, we use a random sample of 80% of the features to learn a linear regression model onto the X-Y coordinates of a 32 x 32 grid, and check performance on the remaining 20%. For methods where the features are of a different resolution, we resize bilinearly.

Method	Segmentation-specific?	Model	mIoU
GroupViT [268]	Yes	modified ViT-S/16	0.53
MaskCLIP [294]	No	ViT-B/16	0.25
Ours	No	ViT-B/16	0.50

Table C.3: **Zero-shot segmentation on PASCAL VOC [71].** Our method is stronger than the MaskCLIP baseline, and competitive with GroupViT, whose architecture is tailored to segmentation.

C.4 More Applications of Per-Image Regions

C.4.1 Adapting CLIP for Open-Vocabulary Semantic Segmentation

As a more interesting case-study than oracle decoding, we assess our regions for zero-shot semantic segmentation on PASCAL VOC [71]. In order to form class decisions, we follow insights from GroupViT [268] and MaskCLIP [294]. First we compute regions on top of CLIP ViT-B/16, then we take the final value vectors from the last attention layer as pixel-wise features, similar to MaskCLIP. We compute region-wise features by averaging pixel-wise features over the regions they correspond to, then compute cosine similarities between these region-wise features and the text embeddings of CLIP, where per-class text embeddings are computed by an average over many different prompts like *"a photo of a {class name}, a picture of a {class name}, ..."*, as is done in GroupViT. Finally we threshold these similarities by a fixed number (0.7), and set all regions to their most similar class, where regions with no similarity greater than the threshold are assigned background. We compare to MaskCLIP [294], a training-free approach, as well as GroupViT [268], which proposes modifications to the original CLIP architecture in order to better suit segmentation.

We see in Table C.3 that, even without a segmentation-specific training objective, we can achieve competitive performance on PASCAL VOC [71], and our region-extraction pipeline aids in segmentation on top of CLIP [195]. We emphasize that this is possible *without any segmentation-specific objectives or additional training*.

Our regions are often contiguous and large in size, while GroupViT's regions contain holes. As a result, the errors that CLIP makes in localizing certain classes may be magnified by our regions. This can be seen in per-class IoU scores in Figure C.5, and examples of CLIP's failure to localize in Figure C.6. Crucially, it appears that CLIP does a poor job localizing particular classes, associating "boat" to any water or beach in the image, "potted plant" and "cow" to ground cover, and "person" to all sorts of human-built objects. Fixing these localization errors in CLIP is out of the scope of our contributions, but could yield improvements to match segmentation-specific methods.

C.4.2 Unsupervised Instance Segmentation

As an additional proof-of-concept, we run experiments on a more difficult task, unsupervised instance segmentation, which requires simultaneously generating object proposals and segmenting salient objects. To benchmark our method, we use the standard COCO 2017 [163] validation split, and follow prior work [255] to report results on both instance segmentation and object detection metrics in a class-agnostic setting. Due to the difficulty of generating instance proposals in a diverse image distribution, recent attempts [254, 257] design heuristic decoding strategies based on the structure of a particular model's features, *e.g.,* the final layer of DINO [34], in order to generate region proposals.

However, we hypothesize that, if the features are informative enough, a simple clustering strategy and generic scoring function should suffice for high-quality instance segmentation. In our implementation, we use K-Means to generate region proposals, and silhouette scores to rank those proposals.

We start by generating initial region proposals by clustering with K-Means on top of the dense features we extract, with K ranging from 2 to 10. To further expand our pool of proposals, we use agglomerative clustering to hierarchically merge spatially adjacent regions with ward linkage.

Naively, we can treat each instance proposal as a binary clustering problem with the foreground and background each as their own cluster, and directly use silhouette scores to rank proposals. However, instances usually take up a relatively small portion of an image making the binary clus-

Method	#Masks	AP^{box}_{50}	AP^{box}	AR^{box}_{100}	AP^{mask}_{50}	AP_{mask}	AR^{mask}_{100}
TokenCut [257]	1	5.2	2.6	5.0	4.9	2.0	4.4
TokenCut [257]	3	4.7	1.7	8.1	3.6	1.2	6.9
MaskCut [255]	3	6.0	2.9	8.1	4.9	2.2	6.9
Ours	13	4.0	1.9	**11.2**	4.0	1.5	**8.2**

Table C.4: **Results of instance segmentation on COCO-val-2017 [163].** Our learned pixel-wise features, with a simple and generic instance segmentation decoding pipeline, significantly outperform baselines in recall, in both object detection and instance segmentation. On the other hand, despite generating many more proposals per image, our method still maintains comparable precision.

tering extremely imbalanced, which significantly harms the scores and ranking.

To this end, instead of treating the complement of foreground masks as background, we subsample the background pixels to create a balanced subset by only preserving the background pixels that are close to the foreground pixels in feature space. We also adopt standard post-processing steps to remove duplicate and extreme-sized segments before producing the final output. Finally, since the silhouette score is in the range $[-1, 1]$, we can use 0 as a threshold to remove low-quality proposals.

We follow the above procedure on top of the features produced by optimizing Eqn. 4.4 over Stable Diffusion's attention layers. We report our results and compare to the current state-of-the-art region proposal methods in Table C.4.

Due to the approximation error in binarizing the affinity matrix for clustering, both TokenCut and MaskCut have trouble yielding diversified samples. By contrast, our learned features contain richer information that allows us to adopt a generic instance grouping pipeline without any post-processing on the features. As we see in Table C.4, this leads us to generate high-quality diversified proposals with better recall in both instance segmentation and object detection metrics, while maintaining comparable precision to prior methods. Qualitative results are available in Figure C.7.

C.5 Code Sources

All experiments are implemented in Python with PyTorch [188]. For Stable Diffusion [204], we use HuggingFace Diffusers [251]. For baselines, we use official numbers, implementations, and model weights, except in the case of MaskCLIP [294], where we reimplement the method due to difficulty in obtaining satisfactory performance.

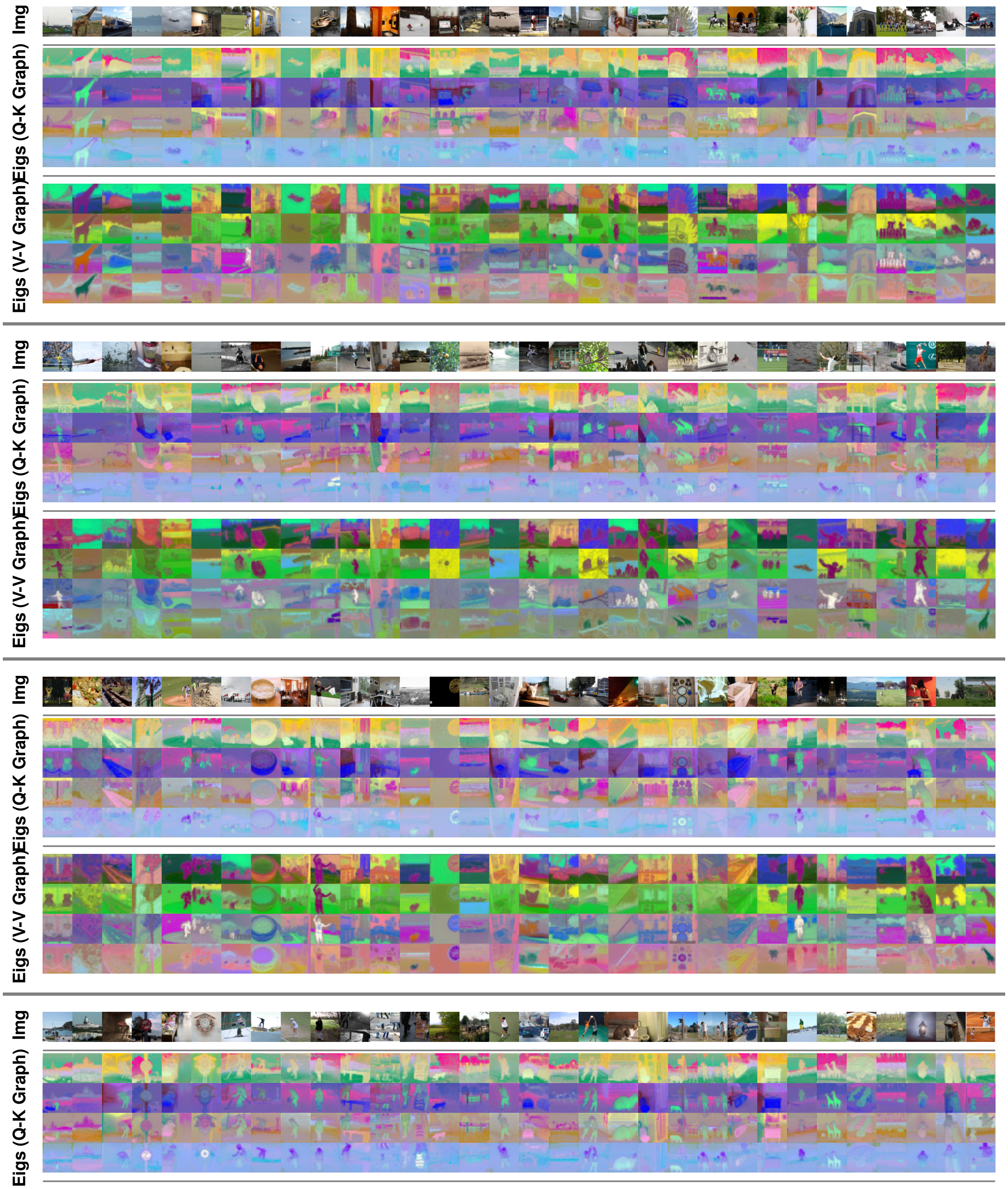


Figure C.1: **More examples of extracted eigenvectors on COCO for both graph choices.** We visualize selected components of $\boldsymbol{X}_{\text{ortho}}$, sorted by decreasing eigenvalue. Three eigenvectors at a time are rendered as RGB images.

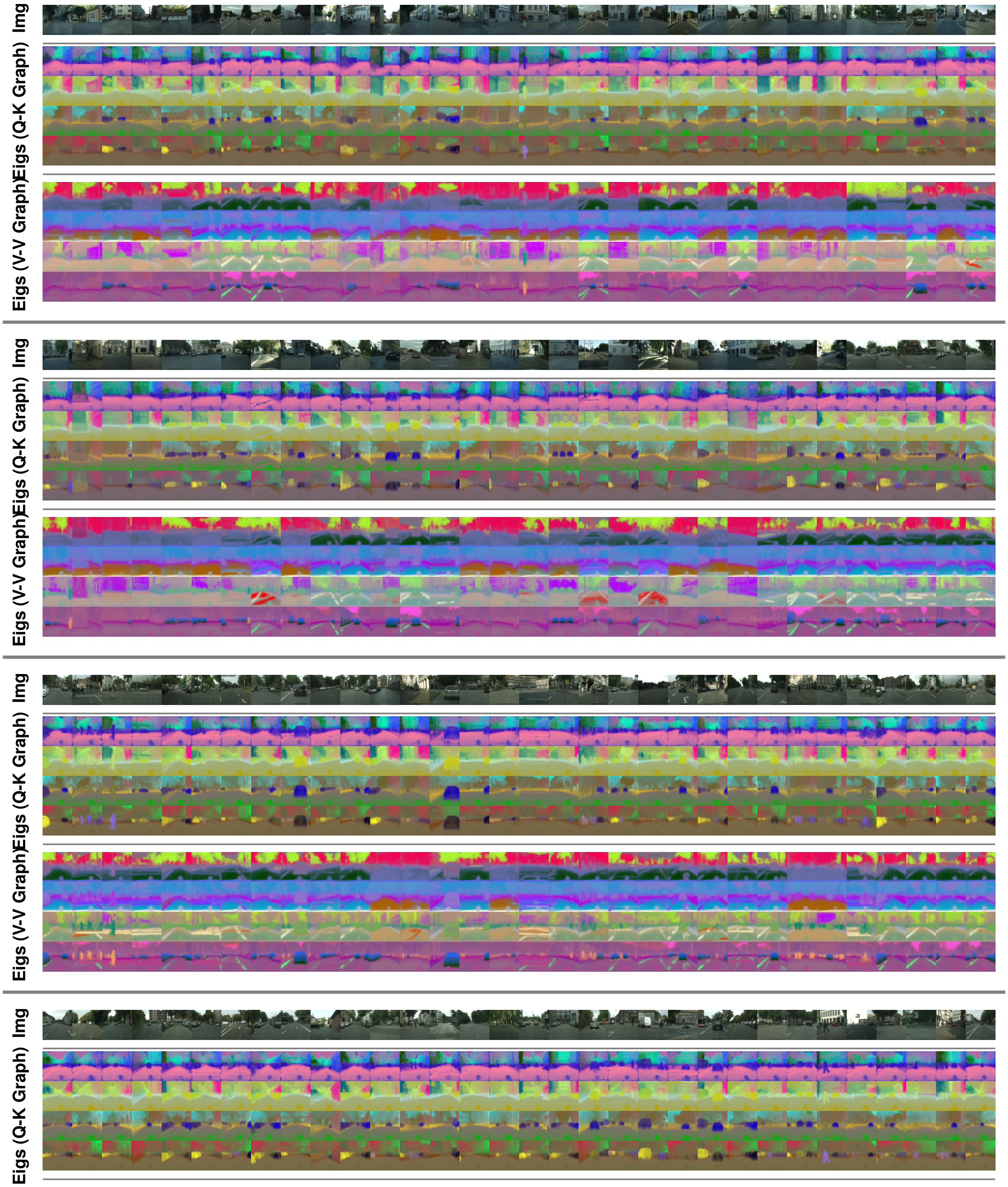


Figure C.2: **More examples of extracted eigenvectors on Cityscapes for both graph choices.** We visualize selected components of $\boldsymbol{X}_{\text{ortho}}$, sorted by decreasing eigenvalue. Three eigenvectors at a time are rendered as RGB images.

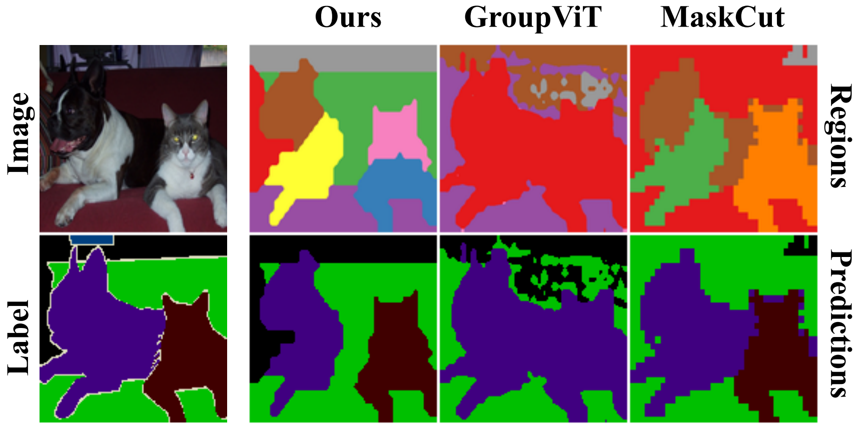


Figure C.3: **Examples of different segmentation methods on PASCAL VOC [71].** All methods besides GroupViT [268] use DINO [34] features or attention. Ours can generate diversified regions while maintaining accurate object borders. In contrast, GroupViT [34] tends to generate a noisy boundary, while MaskCut [255] can miss subtle boundaries.

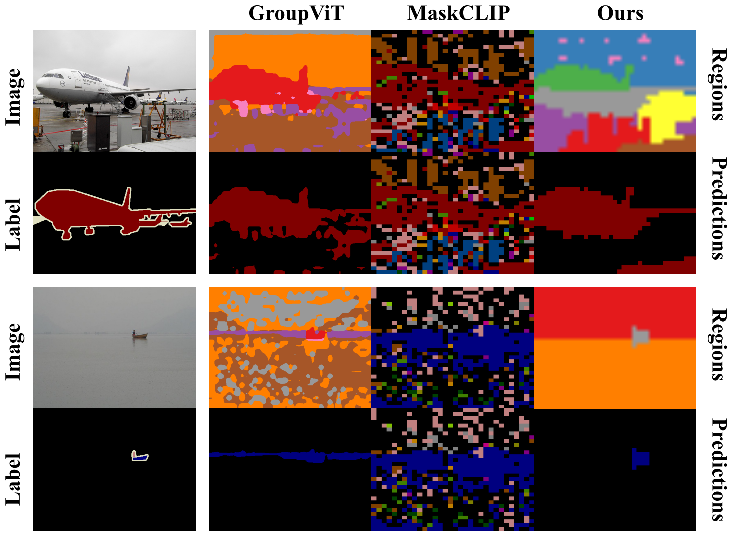


Figure C.4: **Examples for different methods on zero-shot semantic segmentation.** Notice the tendency of GroupViT [268] and MaskCLIP [294] to break up objects, and the eagerness of MaskCLIP to cover the image. On the airplane image we perform slightly worse than GroupViT but our regions have more spatially coherent structures. On the boat image our method has better performance and can even separate water and sky, though the gap between their pixel values is almost imperceptible.

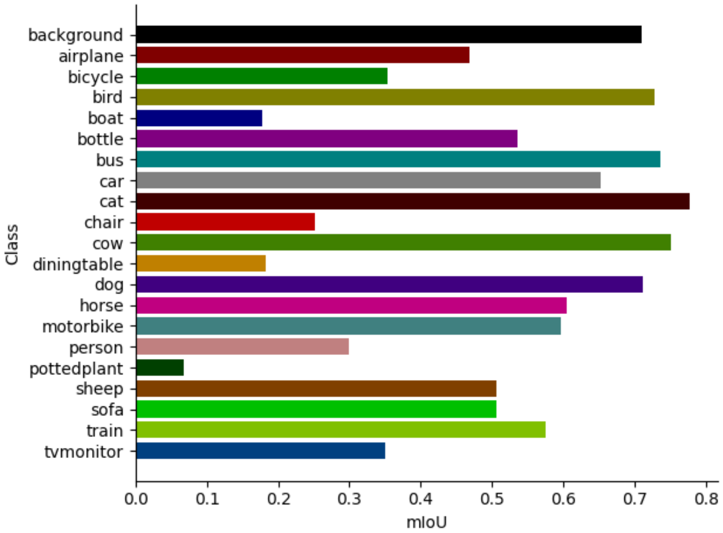


Figure C.5: **Per-class mIoUs on PASCAL VOC.** Errors are pronounced in a few particular classes, like “boat”, “potted plant”, and “dining table,” which are primarily due to localization issues with CLIP.

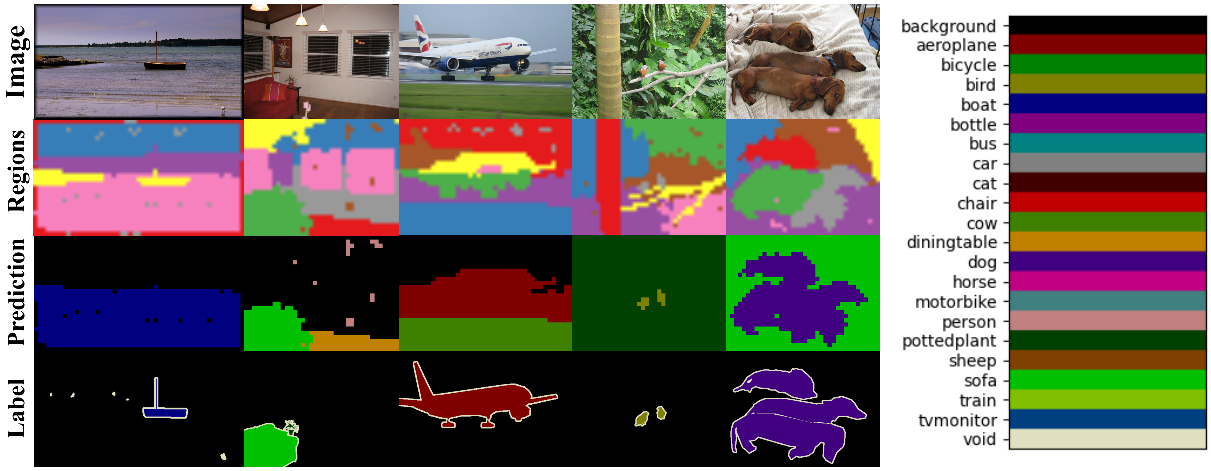


Figure C.6: **Examples of segmentation failures.** From the regions we see that most object are correctly segmented and classified, but CLIP fails on the background. From left to right, water is classified as “boat,” hardwood floor is classified as “dining table,” runway grass is classified as “cow,” forest foliage is classified as “potted plant,” and bedding is classified as “sofa.” This persists across threshold values, as the CLIP similarities are very high. Refining CLIP’s localization ability can close much of the gap to oracle decoding.

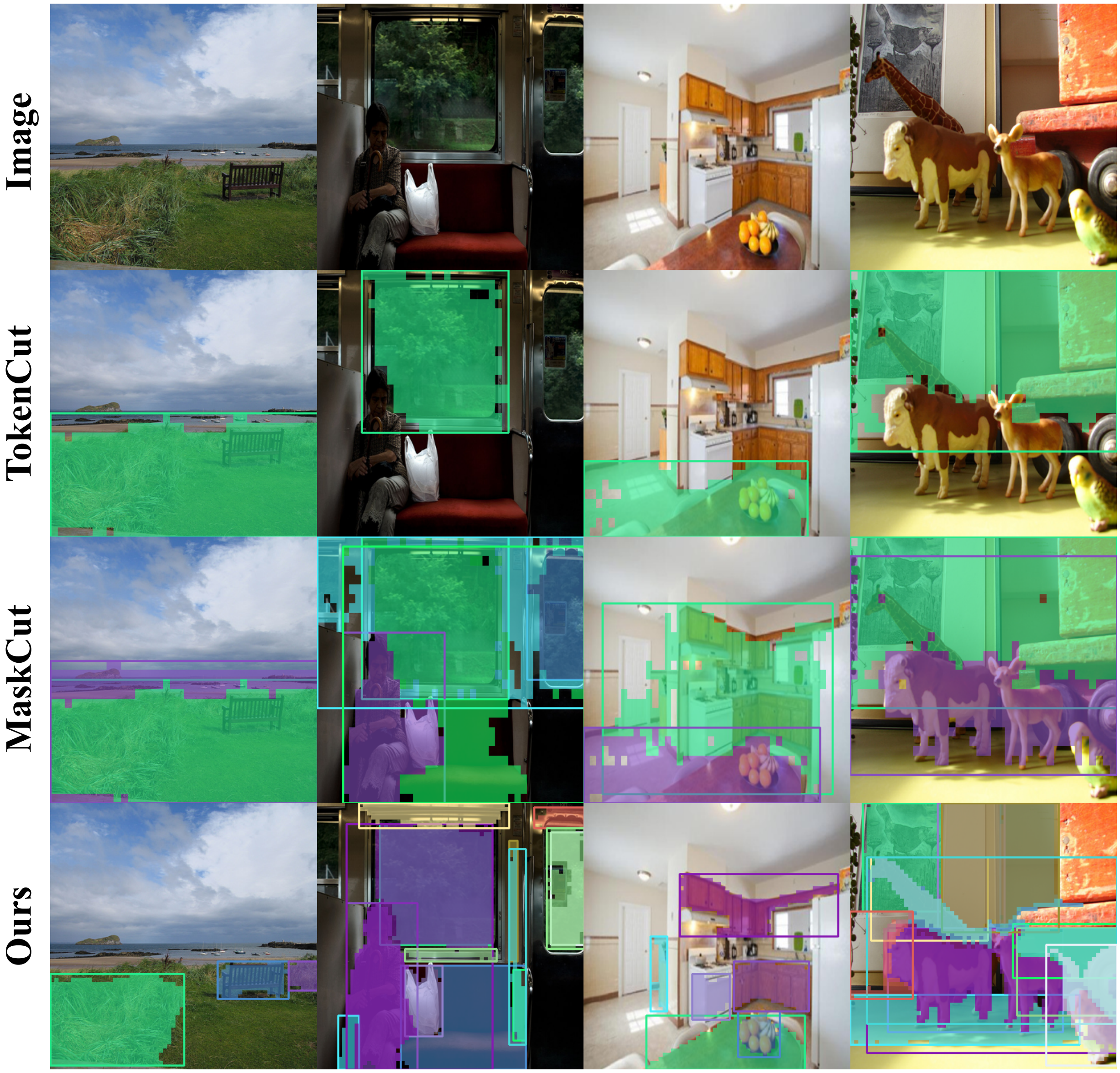
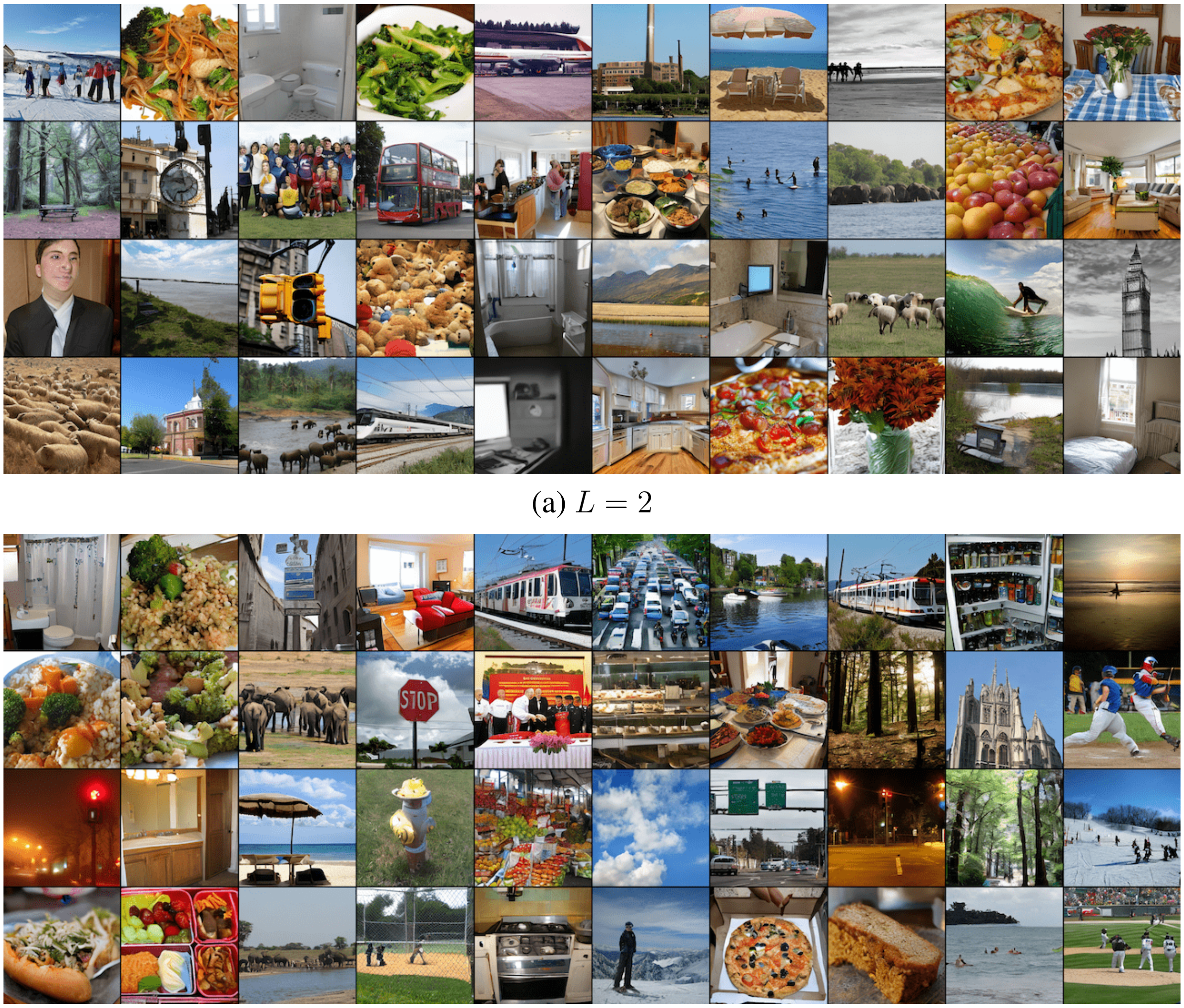


Figure C.7: **Examples of different methods on instance segmentation.** As described by the original authors, TokenCut [257] can only generate a single object proposal, and MaskCut [255] is limited as well. Our method shows better localization results and scales to many instances.

APPENDIX D

NESTED DIFFUSION MODELS USING HIERARCHICAL LATENT PRIORS



(a) $L = 2$

(b) $L = 3$

Figure D.1: **Visualization of text-to-image generation on COCO-2014.** We present example images generated by hierarchical diffusion models of 2 and 3 levels.

D.1 Further comparison

Table D.1 presents a comparison of model parameters and computational complexity for models with varying depths L. In contrast to the baseline model 1^*, whose computational complexity

L	1	2	3	4	5	1^*
GFlops	26.82	27.13	28.17	29.61	33.98	35.11
Param(M)	104	208	313	417	523	135
FID ↓	31.13	16.52	15.50	13.87	**9.87**	19.74

Table D.1: **Comparison of multiple model configurations over model depth L.** Unlike the baseline diffusion model ($L = 1^*$) whose computational complexity (GFlops) grows linearly with model parameters, our efficient hierarchical design only yields minimal GFlops growth with deeper models but achieves much better image quality than the baseline model with the same GFlops.

scales linearly with the number of parameters, our efficient hierarchical design incurs only a modest computational overhead for deeper models. Under a comparable computational budget, our model with $L = 5$ demonstrates significantly better performance than 1^*.

D.2 Derivation of formulas

Derivation for $\mathcal{L}_{\text{ELBO}}$ (Eqn. 5.2). Let $\boldsymbol{x} = \boldsymbol{z}_1$ be the observed data and $\boldsymbol{z}_2, \boldsymbol{z}_3, \ldots, \boldsymbol{z}_L$ be the latent variables with $\boldsymbol{z}_{>l} := \{\boldsymbol{z}_m\}_{m=l+1}^{L}$. We assume the joint distribution of data and latent variables can be modeled as follows:

$$p_\theta(\boldsymbol{x}, \boldsymbol{z}_{>1}) = p_\theta(\boldsymbol{x}|\boldsymbol{z}_{>1}) \prod_{l=2}^{L-1} p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{>l}) p_\theta(\boldsymbol{z}_L), \tag{D.1}$$

with the corresponding posterior written as:

$$q(\boldsymbol{z}_{>1}|\boldsymbol{x}) = q(\boldsymbol{z}_L|\boldsymbol{x}) \prod_{l=2}^{L-1} q(\boldsymbol{z}_l|\boldsymbol{z}_{>l}, \boldsymbol{x}). \tag{D.2}$$

For the derivation of the ELBO, we proceed in a similar way as Vahdat and Kautz [245], Pervez and Gavves [191], Takida et al. [236] by relying on Jensen's equality:

$$\begin{aligned}
\log p_\theta(\boldsymbol{x}) &= \log \int p_\theta(\boldsymbol{x}, \boldsymbol{z}_{>1}) \mathrm{d}\boldsymbol{z}_{>1} && \text{(D.3)} \\
&= \log \int q(\boldsymbol{z}_{>1}|\boldsymbol{x}) \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z}_{>1})}{q(\boldsymbol{z}_{>1}|\boldsymbol{x})} \mathrm{d}\boldsymbol{z}_{>1} && \text{(D.4)} \\
&\geq \mathbb{E}_{q(\boldsymbol{z}_{>1}|\boldsymbol{x})} \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z}_{>1})}{q(\boldsymbol{z}_{>1}|\boldsymbol{x})} && \text{(D.5)} \\
&\equiv \text{ELBO}. && \text{(D.6)}
\end{aligned}$$

By plugging in Eqn. D.1 and Eqn. D.2:

$$\begin{aligned}
\text{ELBO} = \mathbb{E}_{q(\boldsymbol{z}_{>1}|\boldsymbol{x})} \Bigg[& \log p_\theta(\boldsymbol{x}|\boldsymbol{z}_{>1}) && \text{(D.7)} \\
& + \sum_{l=2}^{L-1} \log \frac{p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{>l})}{q(\boldsymbol{z}_l|\boldsymbol{z}_{>l}, \boldsymbol{x})} + \log \frac{p_\theta(\boldsymbol{z}_L)}{q(\boldsymbol{z}_L|\boldsymbol{x})} \Bigg] \\
= \mathbb{E}_{q(\boldsymbol{z}_{>1}|\boldsymbol{x})} & \log p_\theta(\boldsymbol{x}|\boldsymbol{z}_{>1}) && \text{(D.8)} \\
& - \sum_{l=2}^{L-1} \mathbb{E}_{q(\boldsymbol{z}_{>l}|\boldsymbol{x})} D_{\mathrm{KL}} \left(q(\boldsymbol{z}_l|\boldsymbol{z}_{>l}, \boldsymbol{x}) | p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{>l}) \right) \\
& - D_{\mathrm{KL}} \left(q(\boldsymbol{z}_L|\boldsymbol{x}) | p_\theta(\boldsymbol{x}_L) \right).
\end{aligned}$$

To incorporate diffusion models to parameterize $p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{>l})$, we further decompose the KL divergence for each level l. Since we utilize a pre-trained encoder that computes each latent variable $\boldsymbol{z}_l$ directly from the observed data $\boldsymbol{x}$ (see Section 5.3.3), we can simplify the conditional posterior distribution by removing the dependence on $\boldsymbol{z}_{>l}$:

$$\begin{aligned}
& - D_{\mathrm{KL}} \left(q(\boldsymbol{z}_l|\boldsymbol{z}_{>l}, \boldsymbol{x}) | p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{>l}) \right) && \text{(D.9)} \\
= & \int q(\boldsymbol{z}_l|\boldsymbol{x}) \Bigg[\log p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{>l}) - \log q(\boldsymbol{z}_l|\boldsymbol{x}) \Bigg] \mathrm{d}\boldsymbol{z}_l. && \text{(D.10)}
\end{aligned}$$

As the posterior q has no learnable parameters θ, then maximizing the negative KL divergence equals maximizing

$$\int q(\boldsymbol{z}_l|\boldsymbol{x}) \log p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{>l}) \mathrm{d}\boldsymbol{z}_l. \tag{D.11}$$

Now assume that the latent variable $\boldsymbol{z}_l$ is modeled through a diffusion process:

$$p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{>l}) = \int p_\theta(\boldsymbol{z}_l^{(0:T)}|\boldsymbol{z}_{>l}) \mathrm{dz}_{\mathrm{l}}^{(1:\mathrm{T})}, \tag{D.12}$$

where $\boldsymbol{z}_l^{(0)} = \boldsymbol{z}_l$, and $\boldsymbol{z}_l^{(t)}$ denotes the noise latent variable at time step $t, \forall t \in \{0, 1, \ldots, T\}$. Then maximizing the likelihood in Eqn. D.11 amounts to maximizing

$$\begin{align} & \int q(\boldsymbol{z}_l^{(0)}|\boldsymbol{x}) \log p_\theta(\boldsymbol{z}_l^{(0)}|\boldsymbol{z}_{>l}) \mathrm{d}\boldsymbol{z}_l^{(0)} \nonumber \\ = & \int \mathrm{d}\boldsymbol{z}_l^{(0)} q(\boldsymbol{z}_l^{(0)}|\boldsymbol{x}) \tag{D.13} \\ & \log \left[\mathrm{d}\boldsymbol{z}_l^{(1:T)} \frac{p_\theta(\boldsymbol{z}_l^{(0:T)}|\boldsymbol{z}_{>l})}{q(\boldsymbol{z}_l^{(1:T)}|\boldsymbol{z}^{(0)}, \boldsymbol{x})} q(\boldsymbol{z}_l^{(1:T)}|\boldsymbol{z}^{(0)}, \boldsymbol{x}) \right] \nonumber \\ \geq & \int \mathrm{d}\boldsymbol{z}_l^{(0:T)} q(\boldsymbol{z}_l^{(0:T)}|\boldsymbol{x}) \tag{D.14} \\ & \log \left[p_\theta(\boldsymbol{z}_l^{(T)}|\boldsymbol{z}_{>l}) \prod_{t=1}^{T} \frac{p_\theta(\boldsymbol{z}_l^{(t-1)}|\boldsymbol{z}_t^{(t)}, \boldsymbol{z}_{>l})}{q(\boldsymbol{z}_t^{(t)}|\boldsymbol{z}_t^{(t-1)}, \boldsymbol{x})} \right] \nonumber \\ \equiv & -\mathcal{L}_l. \tag{D.15} \end{align}$$

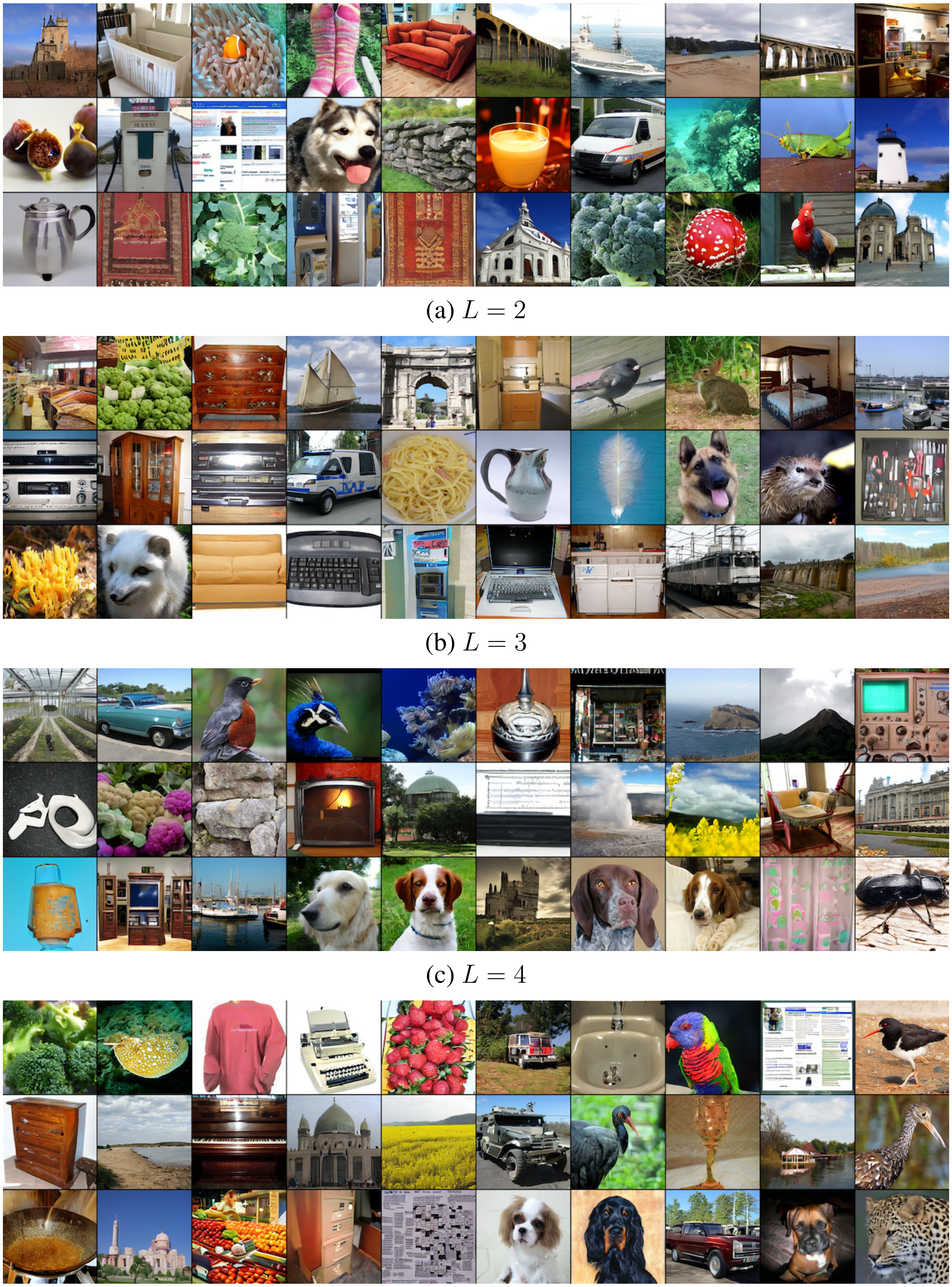
Following the derivation in Sohl-Dickstein et al. [224], the loss at each level l can be further reduced as

$$\begin{align} \mathcal{L}_l \leq & \quad \textstyle\sum_t \int \mathrm{d}\boldsymbol{z}_l^{(0)} \boldsymbol{z}_l^{(t)} q(\boldsymbol{z}_l^{(0)}, \boldsymbol{z}_l^{(t)}) \tag{D.16} \\ & D_{\mathrm{KL}} \left(q(\boldsymbol{z}_l^{(t-1)}|\boldsymbol{z}_l^{(t)}, \boldsymbol{z}_l, \boldsymbol{x}) \| p_\theta(\boldsymbol{z}_l^{(t-1)}|\boldsymbol{z}_l^{(t)}, \boldsymbol{z}_{>l}) \right). \nonumber \end{align}$$

By plugging this reduced form of the loss at each level into Eqn. D.8, we arrive at Eqn. 5.2.

D.3 Qualitative evaluation

Additional visualizations of images generated by our model at different depths are provided: Fig. D.1 illustrates text-to-image generation on the COCO-2014 dataset, Fig. D.2 displays conditional generation on ImageNet-1k, and Fig. D.3 displays unconditional generation on ImageNet-1k.



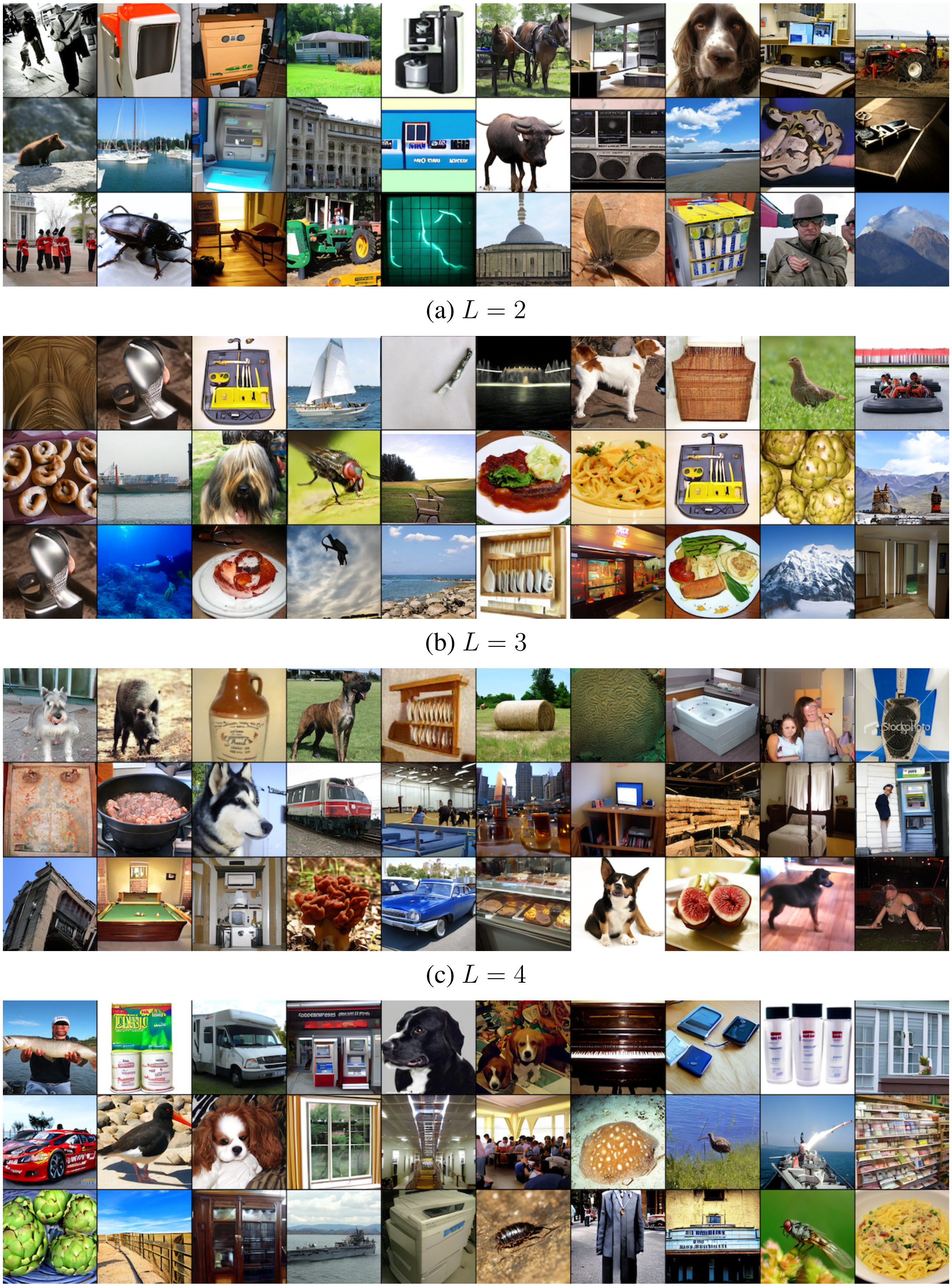
(a) $L = 2$

(b) $L = 3$

(c) $L = 4$

(d) $L = 5$

Figure D.2: **Visualization of conditional image generation on ImageNet-1K.** We present example images generated by hierarchical diffusion models containing from 2 to 5 levels.



(a) $L = 2$

(b) $L = 3$

(c) $L = 4$

(d) $L = 5$

Figure D.3: **Visualization of unconditional image generation on ImageNet-1K.** More example images generated by hierarchical diffusion models containing from 2 to 5 levels.