Unsupervised Learning with Diffusion Models

Wang Jiankun

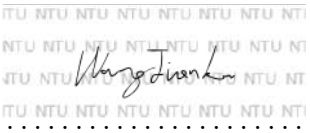
School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Master of Engineering

2023

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.



14/11/2023

Date

Wang Jiankun

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

14/11/2023

Date

Liu Weichen

A/P Liu Weichen

Authorship Attribution Statement

This thesis does not contain any materials from papers published in peer-reviewed journals or from papers accepted at conferences in which I am listed as an author.

14/11/2023

.....

Date

.....

Wang Jiankun

Acknowledgements

Firstly, I would like to express my gratitude to my supervisor Prof. Weichen Liu for his guidance and support. He treats students with great tolerance, patience, and respect for their ideas. During times when my research progress was not smooth, He provided me with tremendous support.

Next, I would like to thank my co-supervisor Prof. Hanwang Zhang. His passion and strict requirements in research helped me develop a strong understanding of research methodologies, encouraging me to conduct meaningful and impactful research.

Furthermore, I am grateful for the help of my friends, Zhongqi Yue, Siyuan Chen, Burak Satar, Haoxin Li, Guyue Hu, Tan Wang, and those who are not listed here. Thank you very much for your assistance and accompany throughout my pursuit of the master degree.

Last but not least, I want to thank my parents for their support and encouragement. I sincerely wish you ongoing good health, happiness, and fulfillment.

Abstract

In computer vision, a key goal is to obtain visual representations that faithfully capture the underlying structure and semantics of the data, encompassing object identities, positions, textures, and lighting conditions. However, existing methods for un-/self-supervised learning (SSL) are restricted to untangling basic augmentation attributes such as rotation and color modification, which constrains their capacity to efficiently modularize the underlying semantics. In the thesis, we propose DiffSiam, a novel SSL framework that incorporates a disentangled representation learning algorithm based on diffusion models. By introducing additional Gaussian noises during the diffusion forward process, DiffSiam collapses samples with similar attributes, intensifying the attribute loss. To compensate, we learn an expanding set of modular features over time, adhering to the reconstruction of the Diffusion Model. This training dynamics biases the learned features towards disentangling diverse semantics, from fine-grained to coarse-grained attributes. Experimental results demonstrate the superior performance of DiffSiam on various classification benchmarks and generative tasks, validating its effectiveness in generating disentangled representations.

Contents

List of Figures

List of Tables

Symbols and Acronyms

Symbols

$\mathcal{X}$	Sample space
$\mathcal{Z}$	Vector space
$\mathcal{Z}_i$	Modular attribute
Φ	Mapping $\mathcal{Z} \to \mathcal{X}$
f	Mapping $\mathcal{X} \to \mathcal{Z}$
f_ω	Encoder as disentangled representation learner parameterized by ω
p_γ	The predictor network in SimSiam parameterized by γ
$\mathbf{x}_0$	Original sample
$\mathbf{x}_t$	Noisy sample after t forward step
$q(\cdot)$	Distribution in the forward process
$p_\theta(\cdot)$	Approximate Distribution in the reverse process
u_θ	θ-parameterized U-Net
g_ϕ	ϕ-parameterized decoder to similuate $\nabla_{\mathbf{x}_t} \log p_\phi\left(\bar{\mathbf{z}}_k \mid \mathbf{x}_t, t\right)$ for guidance
$\hat{\mathbf{x}}_0$	Reconstructed $\mathbf{x}_0$
$\mathbf{z}$	The feature vector inferred from the encoder f_ω
$\mathbf{z}_i$	i-th modular subset of $\mathbf{z}$
$\bar{\mathbf{z}}_k$	$[\mathbf{z}_1, \ldots, \mathbf{z}_k, 0, \ldots, 0]$
n	Number of subsets of $\mathbf{z}$
T	Total time-steps
$\beta_1, \ldots, \beta_T$	Variance schedule
α_t	$1 - \beta_t$
$\bar{\alpha}_t$	$\prod_{s=1}^{t} \alpha_s$

Acronyms

SSL	Self-supervised Learning
DM	Denoising Diffusion Probabilistic Model
slerp	Spherical linear interpolation
lerp	Linear interpolation

Chapter 1

Introduction

1.1 Background and Motivation

In computer vision, a key goal is to obtain visual representations that faithfully capture the underlying structure and semantics of the data, encompassing object identities, positions, textures, and lighting conditions. As emphasized in various studies [3–5], this representation should enable not only effective sample inference tasks such as image classification but also the generation of counterfactual instances, such as synthesizing images with unseen combinations of attributes [6]. Un-/self-supervised learning (SSL) has emerged as a promising approach for learning powerful visual representations. The principle of SSL involves initially training a universal feature representation, commonly termed a network backbone, by training on pretext tasks like contrastive learning [7, 8]. While SSL methods have impressed with their strong performance on typical tests and even outperformed supervised approaches in different practical applications, it's important to note that the features they learn are primarily drawn from manually designed augmentations [4], like altering colors or rotating images. This focus on specific augmentation limits their potential to grasp a wider array of diverse representations.

To address this limitation and push the boundaries of self-supervised learning, we propose a novel SSL framework called DiffSiam, which incorporates disentangled representation learning using diffusion models.

Diffusion Probabilistic Models (DM) [9, 10] have proven to be highly effective in preserving all underlying attributes for faithful generation [11, 12], even allowing for the synthesis of novel attribute combinations through textual control. This suggests that DMs successfully capture the modular nature of hidden attributes [13]. It motivates us to study the potential algorithm to extract representations from DM. However, since DMs lack explicit encoders that transform samples into feature vectors, the conventional encoder-decoder feature learning paradigm, which relies on reconstruction, is no longer applicable [14–16].

Is it possible to extract the knowledge about the modular attributes from a generative Denoising Diffusion Probabilistic Model (DM) into a discretized feature vector? Our key observation is the relationship between diffusion time-steps and the loss of modular attributes. This insight arises from analyzing the loss of attributes throughout the diffusion process. Initially, we have an original image sample, $\mathbf{x}_0$, encompassing all information about the visual properties, or attributes, of objects in the image. As we initiate the diffusion process, Gaussian noise $\mathcal{N}(0, \mathbf{I})$ is incrementally added to this sample across T diffusion time-steps. This gradual addition of noise begins to remove some of the image attributes, leading to a series of increasingly noisy samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$. Each noisy sample at a middle time-step $\mathbf{x}_t$ (where $1 \leq t \leq T$) aligns with a Gaussian distribution $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I})$. As the diffusion process ends up with x_T, the image transforms into standard Gaussian noise, which signifies the complete loss of visual attributes of the original image. This inherent relationship between diffusion time-steps and the loss of modular attributes could serve as an inductive bias for self-supervised learning.

The proposed DiffSiam introduces additional Gaussian noises during the diffusion forward process, intensifying the attribute loss. To compensate, we learn an expanding set of modular features over time, adhering to the reconstruction of the Diffusion Model. This training dynamics biases the learned features towards disentangling diverse semantics, from fine-grained to coarse-grained attributes.

Drawing inspiration from previous works [2, 17], we take advantage of a pretrained DM to help representation learning. In these models, an encoder f_ω paramaterized by ω is trained atop DM using reconstruction loss to process $\mathbf{x}_0$. To achieve accurate reconstruction, the feature vector $\mathbf{z} = f_\omega(\mathbf{x}_0)$ has to compensate for the loss of attributes, thereby forming associations between $\mathbf{z}$ itself and semantic attributes.

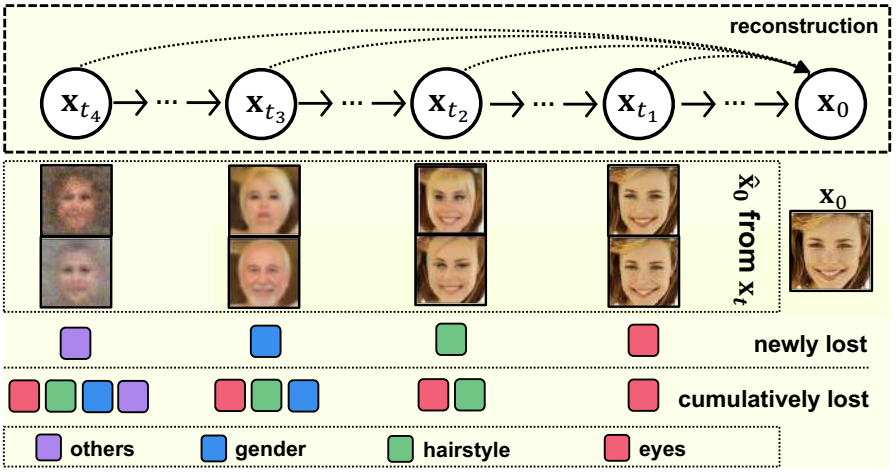


FIGURE 1.1: Illustration of the lost attributes in the diffusion forward process. The DM is pretrained on CelebA [1], from which we randomly draw an original image $\mathbf{x}_0$ (a woman's face). Applying diffusion forward process, we get noisy samples at various timestep ($t_4 > t_3 > t_2 > t_1$). The reconstructed $\mathbf{x}_0$, denoted as $\hat{\mathbf{x}}_0$, from noisy samples reflects the lost attributes in different time-step.

Moreover, we observe a pattern while losing attributes in the diffusion process. To illustrate, in Figure 1.1, fine-grained attributes (like eyes) get lost at earlier diffusion time-steps, whereas coarse-grained attributes (like hairstyle, gender) at later time-steps. Our framework is designed to disentangle these attributes at varying levels of granularity, by utilizing distinct subsets of the feature vector for reconstruction. Granularity is quantified by the amount of pixel-level changes. To this end, feature vector $\mathbf{z}$ is divided into disjoint subsets, with each subset dedicated to learning the attributes lost within a particular range of time-steps. This methodology enables the encoding of an image's attributes into a modular vector space.

Our experimental results demonstrate the superiority of DiffSiam on various classification benchmarks and generative tasks, outperforming baseline methods in several settings. Furthermore, DiffSiam enables counterfactual generation on datasets like FFHQ and Bedroom, validating its effectiveness in producing disentangled representations.

1.2 Contributions

Our contributions can be stated as follows:

- We propose to leverage the inductive bias between the diffusion time-steps and the underlying hidden attributes for unsupervised disentangled representation learning.
- We present a novel SSL framework called DiffSiam, incorporating the diffusion models into standard SSL network, SimSiam [18].
- Through extensive demonstrations, we show that our algorithm outperforms baseline approaches on various classification benchmarks and generates samples of significantly higher quality in tasks like counterfactual generation and attribute manipulation.

1.3 Outline of the Thesis

The structure of the thesis is as follows:

Chapter 1 introduces the background and motivation of the thesis: it recognizes the limitation of un-/self-supervised representation learning and briefly discusses how the proposed approach based on the inductive bias inherent in diffusion models may help achieve disentanglement.

Chapter 2 systematically reviews the concepts, techniques and tasks related to our proposed methods: the approaches and evaluation protocols of self-supervised learning, the definitions and techniques of disentangled representation learning and the works related to representation learning with diffusion models.

Chapter 3 starts from the formulation of some techniques our approach will use. Building upon this foundation, we present our unsupervised disentangled representation learning approach, named DiffSiam.

Chapter 4 presents the experimental setup and presents a comparative analysis of the results obtained using our approach and baseline methods. We specifically focus on evaluating the performance of our approach in tasks like many-shot classification, few-shot classification, counterfactual generation and attribute manipulation.

Chapter 5 summarizes the contributions of our work. Furthermore, we identify potential future research directions for advancing unsupervised representation learning with diffusion models, emphasizing enhancing disentangled properties and exploring improved diffusion frameworks.

Chapter 2

Literature Review

In this chapter, we review the techniques of self-supervised learning (Section 2.1), disentangled representation learning (Section 2.2) and diffusion models (Section 2.3) related to our proposed methods.

2.1 Self-Supervised Learning

In real-world scenarios, the limited availability of labeled data poses a significant challenge for various tasks, such as medical image segmentation. Un-/self-supervised learning (SSL) of image representations offers a promising solution to address this issue. By formulating pretext tasks based on unlabeled inputs, SSL learns general features about the data, which can then be leveraged for downstream tasks like classification or segmentation. Early progress in SSL explored diverse pretext tasks, including colorization of greyscale images [19], solving a jigsaw puzzle [20], predicting the ego-motion of a camera from multiple frames [21]. In the contemporary landscape, SSL methods have seen substantial advancements, fueled by large datasets like ImageNet [22] and high-memory GPUs. These methods primarily focus on maintaining invariance to heavy data augmentations, allowing them to extract robust and informative representations.

We will present an overview of popular SSL approaches (Section 2.1.1), followed by an exploration of evaluation protocols employed to assess their performance (Section 2.1.2). Finally, we summarize our review with a discussion on the limitations of current SSL approaches and their relevance to our research study (Section 2.1.3).

2.1.1 Approaches

According to whether there are negative samples, the SSL method can be roughly divided into two categories: contrastive and non-contrastive ones.

Contrastive methods

Contrastive methods bring together embeddings of different views (positive samples) of the same image (anchor) while pushing away the embeddings from different images (negative samples). The major challenge is to deal with the large number of negative samples. To model the underlying continuous feature space, we need to model the space with negative samples as much as possible under the constraint of storage and computational cost to process the large number of samples at the same time. Thus, a series of contrastive approaches were proposed. They either use external data structure [23, 24], clustering, or end-to-end learning with large batch size [8] to deal with the substantial need for negatives.

[23] proposes to store feature representations of all samples in a dictionary, or memory bank. In order to tackle the difficulty of computing the similarity to all the samples in the memory bank, they adapt noise-contrastive estimation (NCE) [25] to the problem, so that they can optimize a part of the memory bank in each training iteration. NCE addresses the multi-class classification challenge by breaking it down into a series of binary classification tasks. In these binary classifications, the goal is to discriminate between the distribution of genuine data and the distribution of fabricated noise. In contrastive SSL, the genuine data is positive samples defined by the pretext tasks. The noise distribution is by contrast the negative sample pairs. The normalizing constant of the data distribution is estimated via Monte Carlo approximation. Due to oscillations caused by random sampling during the learning process, [23] introduces a momentum term into the objective function to promote smoother training dynamics.

While the memory bank introduced by [23] is capable of accommodating a large dictionary size, it has the drawback of containing keys from various encoders across the previous epoch, resulting in inconsistency. To address this issue, MoCo [24] introduces a momentum encoder as an alternative regularization technique. It employs a high momentum, typically set to a default value of 0.999. The higher momentum works better to ensure the consistency of the representations of keys. Additionally, it employs a queue structure to store multiple negative samples from preceding batches. The queue size is much smaller than the dataset, typically set to 65536 for the 1.2 million ImageNet dataset. Leveraging the properties of the queue, the current mini-batch is enqueued into the dictionary, while the oldest mini-batch in the queue is removed. The keys encoded from the oldest mini-batch are the least consistent with the most recent ones, making them outdated.

SimCLR [8] takes a different approach from constructing offline negative sample storage. Instead, it utilizes in-batch negatives during end-to-end optimization. The two views of the same image are generated through a combination of augmentations like cropping, resizing, and color jittering. Following encoding using a shared encoder, SimCLR utilizes a projector to convert the initial embeddings into an alternate space, where a contrastive loss is applied between the distinct views. Importantly, performance improvements in subsequent tasks have been attributed to the projection following the feature extractor. And the good performance also relies on the combination of random cropping and random color distortion.

Further reducing the computational cost of contrasting a large number of pairwise features, SwAV [26] proposes to contrast between prototypes, or clusters. It first forms target codes by mapping representations from two augmented views of the input to the prototypes. Then the two target codes are predicted to each other. Moreover, SwAV proposes an effective augmentation technique named multi-crop without imposing additional memory or computational burdens during training. Multi-crop involves incorporating low-resolution crops alongside higher-resolution crops of the image, enabling the model to capture both local and global features. Notably, SwAV, when trained with 3000 prototypes, demonstrates competitive performance with SimCLR [8] on the ImageNet dataset.

Non-contrastive methods

While negative pairs in contrastive SSL methods provide constraints to prevent mode collapse, they often need careful treatments, such as large batch sizes [8, 27], memory banks [7, 23] or some hard-negative mining algorithms [28, 29].

Non-contrastive methods are proposed to remove the need for explicit negative pairs. They either rely on architectural design [18, 30] or regularization of the covariance of the embeddings [31–33] to avoid the trivial constant solution to the SSL without negatives.

BYOL [30] reformulates the discrimination problem between similar and dissimilar pairs into a prediction problem. It trains two branches of networks, using one branch (online) to predict the other branch's (target) feature of another augmented view of the same image. The training objective does not explicitly include a negative term that would encourage its features to spread apart. However, the representation does not collapse. It is hypothesized in BYOL that the avoidance is due to the combination effect of the predictor layer and the momentum encoder (online branch of the network).

SimSiam [18] also removes the negative sample pairs and it further challenges the necessities of the momentum encoder shown to be necessary in [30]. Based on empirical evidence, SimSiam establishes that the avoidance of mode collapse is brought by the utilization of the stop-gradient mechanism.

Barlow Twins [31] proposes an objective function to avoid the mode collapse. It calculates the cross-correlation matrix between the embeddings of two identical networks, both fed with distinct distorted iterations of a batch of samples. The goal is to converge this matrix toward the identity matrix. This process enhances the resemblance between embedding vectors of the distorted sample versions, concurrently minimizing redundant elements within these vectors.

2.1.2 Evaluations

The evaluation of self-supervised learning (SSL) methods is crucial to understanding their performance and effectiveness in learning meaningful representations without explicit labels. While image classification has been the traditional benchmark for evaluating SSL, recent advances have explored other vision tasks like object detection and semantic segmentation. One of the most popular evaluation methods

for SSL is linear-probing evaluation, introduced by [19, 34]. This strategy encompasses training a linear classifier atop the pre-trained feature representations, offering a straightforward yet impactful method to evaluate the learned features' quality. The frozen backbone is followed by a linear classifier in the form of a final layer. The parameters of this classifier are fine-tuned over a number of epochs. The success of this method lies in its efficiency, ability to mimic practical use cases, and reliance on high-quality representations.

Despite classification tasks often serving as the evaluation protocol in SSL, there has been increasing interest in exploring the benchmarks like semantic segmentation and object detection [32, 35, 36]. The main challenge lies in the lack of standardized evaluation protocols for these scenarios. There are varied methods available, including adjusting the encoder for downstream tasks or utilizing the encoder as a feature extractor. Creating a unified assessment framework for these objectives within the domain of self-supervised learning undergoes active research.

An alternative evaluation method involves analyzing the information contained within the learned representations. One approach is to employ a decoder that can map the representations back to pixel space. While some methods [37] come equipped with a specific decoder for visual analysis, many SSL methods do not. [38] deals with it by putting forward a conditional generative diffusion model, utilizing SSL representations as conditioning factors. This allows the examination of information that remains constant across different generated samples and information that exhibits variance due to the generative model's stochasticity. By interpreting the variance in generated samples, valuable insights into the content of the representations can be gained. For instance, if a representation encodes fine-grained pixel information, the generative model should reconstruct images with minimal variation. On the other hand, if the representation primarily encodes class information, generated samples will maintain the object's class while exhibiting variations in background, context, or color.

In conclusion, evaluating self-supervised learning methods involves a range of techniques, including linear-probing evaluation for image classification, exploring other vision tasks, and utilizing generative evaluation for visual analysis. Standardization of evaluation protocols remains to be explored for further understanding the quality and utility of learned representations.

2.1.3 Discussions

To wrap up, most self-supervised learning (SSL) recent approaches are mainly composed of contrastive methods and non-contrastive ones. Evaluating SSL methods involves a range of techniques, including linear-probing evaluation for image classification, exploring other vision tasks, and utilizing generative evaluation for visual analysis. Open research questions remain regarding the theoretical view of SSL, generalization guarantees, and robustness to adversarial attacks. Current methods are only capable of separating basic augmentation elements, making them struggle to effectively exploit other diverse semantics. Our proposed method integrates disentangled representation within SSL to demonstrate the constraints of existing approaches. The simplicity of SimSiam [18] and its ability to achieve competitive performance with fewer computational resources make it an attractive self-supervised learning approach. Therefore, we choose it as the baseline approach in our study.

2.2 Disentangled Representation Learning

Disentanglement is a challenging objective in the domain of representation learning, developing models that can exploit the underlying generative and interpretable factors within data. By achieving this, models are expected to offer valuable contributions to downstream tasks. For instance, in the context of a disentangled representation of a handwritten digit image, individual dimensions may correspond to distinct attributes, such as colors, digit types, widths, slants, etc. Leveraging such knowledge about the attributes and concepts is able to improve the robustness and precision of downstream tasks, including classification and recognition.

This section reviews disentangled representation learning, which is about the underlying modular hidden attributes in data generation. We first present the definitions and properties of disentanglement, as well as the main approaches employed in unsupervised disentangled representation learning. Finally, we conclude the challenges faced in this field and emphasize the connection between our work and unsupervised disentangled representation learning, showcasing what our research contributes.

2.2.1 Definitions and Properties

Initially, researchers are limited to disentangling the variational factors inside some toy datasets with known ground truth generative processes. It was unclear how to extend these notions to more complex real-world scenarios.

However, [39] sheds light on the concept of disentangled representations and emphasizes three crucial properties they should exhibit. By striving to achieve modularity, compactness, and explicitness, the pursuit of unsupervised learning aimed to automatically discover and disentangle the generative factors of the samples.

- Modularity: each latent dimension is confined to encode no more than a single generative factor.
- Compactness: each generative factor is supposed to be encoded by some latent dimension.
- Explicitness: the latent representation must encompass all generative factors and enable their decoding through linear transformations, wherein linearity imposes a stricter requirement.

Building on these fundamental insights, the **group-theoretic viewpoint** emerged as a promising direction in understanding disentangled representations. It was inspired by the notion that various transformations observed in the real world could be perceived as symmetry transformations, thereby forming symmetry groups. In this context, Higgins [5] introduced a definition that explicitly connected disentangled representations to the symmetry group. Such a definition requires the group action on the symmetry group to be equivariant to the action in the space where disentangled representation lies in. In pursuit of practical applications, [4] presented an iterative algorithm to learn the representation through invariant risk minimization, accompanied by a theoretical assurance of convergence to a fully disentangled representation. Simultaneously, [40] involves encoding data variations with groups, a structure that not only can equivariantly represent variations but can also be adaptively optimized to preserve the properties of data variations.

Subsequently, researchers [41] sought to formalize the disentangled representation properties from an **information-theoretic viewpoint**, aiming to design appropriate training objectives and evaluation metrics. By evaluating the disentangled representations' informativeness, separability, and interpretability concerning ground truth factors, they endeavored to quantify the extent to which a representation achieves full disentanglement.

From an information-theoretic viewpoint, the concept of "fully disentangled" is defined concerning a human-interpretable factor y_k rewriting representation z_i to be fully separable (from $z_{\neq i}$) and fully interpretable w.r.t y_k. Mathematically, this condition is expressed as: $I(z_i, z_{\neq i}) = 0$ and $I(z_i, y_k) = H(z_i, y_k)$. They quantify three important properties:

- *Informativeness* constitutes a crucial aspect of disentanglement, referring to the representation's ability to faithfully capture and represent the underlying data x. It is formulated as the mutual information between z_i and x.

$$I(x, z_i) = \int_x \int_z p_{\mathcal{D}}(x) q(z_i|x) \log \frac{q(z_i|x)}{q_{z_i}} dz dx. \tag{2.1}$$

- *Separability*, on the other hand, deals with the independence between two representations, two representations z_i, z_j are independent with respect to the data x, is measured by the multivariate mutual information.

$$I(x, z_i, z_j) = 0. \tag{2.2}$$

  Futher, z_i and z_j are deemed independent if and only if $I(z_i, z_j) = 0$.

- *Interpretability.* However, obtaining informative and independent representations does not necessarily ensure human interpretability, as highlighted by [42]. To achieve interpretability, it is essential to establish "full interpretability" concerning a predetermined set of concepts y. This requires $I(z_i, y_k) = H(z_i) = H(y_k)$.

The information-theoretic perspective has provided a framework for evaluating disentanglement quality in terms of informativeness, separability, and interpretability.

Taking advantage of them, approaches using VAEs and GANs have been extensively explored, showcasing their potential in effectively disentangling representations (Section 2.1.2).

2.2.2 Approaches

Here, we provide an overview of the approaches used with VAEs and GANs to tackle disentanglement.

VAE-based Approaches

Variational Autoencoders [43] was found the ability to learn disentangled representation on simple datasets, particularly on simple datasets like MNIST [44]. Their original objective is to maximize the (variational) evidence lower bound (ELBO), consisting of the conditional logarithmic likelihood responsible for decoding the data from the latents and the KL divergence reflects the distance between the variational posterior distribution and the prior distribution of the latents.

To avoid posterior collapse, researchers have devised several additional regularizers that can be incorporated into the original VAE objective. This has given rise to a collection of VAE-based approaches [45–49].

VAE's disentanglement on toy datasets comes from the KL term that imposes independent constraints on the latents. To control the independence constraints, β-VAE [45] introduces the loss weight β as the penalty coefficient before the KL term. A larger penalty tends to increase the disentanglement while degenerating the reconstruction quality. InfoVAE [46] incorporates mutual information as a regularization term to guide the learning process.

GAN-based Approaches

In contrast to VAEs, GANs [50] take an adversarial learning approach to map data samples into a structured and interpretable feature space. GANs consist of two neural networks: a generator that creates synthetic data to mimic real samples, and a discriminator that distinguishes between real data and generated data. Through a two-player minimax game, GANs aim to find a balance where the generator produces realistic samples that the discriminator finds hard to tell the real ones from the fake ones apart.

Unsupervised disentanglement based on GANs has seen advancements through approaches like [51–54].

InfoGAN [51] introduces additional latent codes to control underlying factors while promoting "informativeness" through optimizing the mutual information. Through this process, it fosters the emergence of meaningful and interpretable latent factors, enhancing the model's ability to capture the underlying data variations effectively. Although InfoGAN has shown advancements in enhancing sample quality, it falls behind its VAE-based counterparts concerning disentanglement scores. InfoGAN-CR [54] incorporates contrastive learning to design a regularizer that encourages distinct changes in the learned representation. Elastic-InfoGAN [52] actively discovers disentangled categorical representations even from imbalanced data.

2.2.3 Challenges

Unsupervised disentangled learning, which is a highly desirable approach for its potential in understanding complex real-world concepts, remains to be explored because it is complex to encode the diverse correlated concepts into separable vectors. [42] proves that without the introduction of inductive biases, there is an infinite number of solutions to disentanglement. These biases involve assumptions about the structure inside datasets or the model. As we delve into the exploration of unsupervised disentanglement, we identify several key challenges that require further investigation, including dealing with real-world datasets, exploring diverse learning paradigms, and investigating alternative models beyond VAEs and GANs.

To date, most VAE-based and GAN-based methods mainly focus on image generation tasks over simple synthetic datasets. The models struggle to disentangle complex scenes. A more intriguing direction involves analyzing and experimenting with more complicated datasets, reflecting the diversity and intricacy of real-world scenarios. Moreover, while existing approaches predominantly center on VAEs and GANs, there is great promise in exploring other potential models, such as diffusion models, which could open new avenues for unsupervised disentangled representation learning. Our work contributes to unveiling one inductive bias by exploring the diffusion time-step and showcasing its efficacy in image generation with complex datasets like FFHQ [55] and Bedroom [56].

2.3 Diffusion Models

Diffusion models attract great attention because they are easy to train and good at generating high-quality and diverse samples. This popularity triggers the applications across diverse fields, such as Text-to-Image generation [12, 57–59], Video generation [60–62], Text-to-3D generation [63–65], and Drug design [66]. We start with an introduction to their fundamental formulation and development in Section 2.3.1, offering readers a thorough grasp of this influential technique. Subsequently, we delve into two applications of diffusion models that are closely connected to our work, specifically focusing on image manipulation and representation learning in Sections 2.3.2 and 2.3.3.

2.3.1 Introduction

Inspired by an idea from non-equilibrium statistical physics, gradually converting a distribution to another by a Markov chain, [9] firstly formalized diffusion models and demonstrated their effectiveness. Their diffusion model consists of two Markov chains. The former is designed to turn the data distribution into a prior distribution, such as a normal Gaussian or an independent binomial distribution). The latter is to learn a Markov chain defined by deep neural networks. Then starting from the prior distribution, this process is able to generate new data points with superb diversity and quality. However, their resulting sample quality is somewhat poor. Several years later, until Denoising Diffusion Probabilistic Models [67] appeared did the diffusion models produce exceptional image samples, with some modifications such as adopting Unet [68] as the backbone, fixed variance schedules, etc. Thanks to the two works, we witnessed an explosion of diffusion-based works thereafter.

2.3.2 Image Manipulation with Diffusion Models

Image manipulation involves generating a target image from a given image and a descriptive text containing instructions for desired changes, such as adding lipstick or glasses to a person's face. This task is increasingly popular for its potential

applications in image editing software. Researchers have explored several categories of approaches to image manipulation using diffusion models.

The first category involves unconditional diffusion models, where conditional information is incorporated either through fine-tuning [69] or classifier guidance [70]. They mainly employ CLIP [71] as the bridge between the descriptive text and generated image. The second class emerged with the introduction of Latent Diffusion [57], which is trained with conditional information but utilizes classifier-free guidance [72]. These approaches employ algorithms such as attention maps [73, 74] or text embeddings [75–77] to manipulate images. Here we will focus on introducing several classic works related to the former mainstream approach, as our work is also based on unconditional diffusion models.

[78, 79] utilize the gradients from CLIP [71] as the classifier to shift the noise along the generation process. By matching noisy images with target text during the reverse process, they manipulate the resulting images without modifying the pre-trained unconditional diffusion models.

DiffusionCLIP [69] proposes a direct fine-tuning approach for diffusion models guided by CLIP [71]. Given an original image x_{ref} (e.g., an expressionless face), its paired textual description y_{ref} (e.g., "face"), and a textual description y_{tar} for the edited image with the desired attribute (e.g., "an angry face"), DiffusionCLIP aims at generating an image satisfying: 1) aligning with the attribute described in y_{tar} (e.g., alter the expression to appear angry). 2) except the attribute in y_{tar}, the generated image x_{tar} should maintain other attributes in the original image x_{ref} (e.g., retain the characteristics of the hair, nose, etc.).

To achieve this, the approach is carried out in two aspects. Firstly, they use the supervision signal from CLIP with two losses. The global loss tries to minimize the cosine distance between x_{tar} and y_{tar} in the CLIP space. The local CLIP directional loss [80] induces the direction between the embeddings of the x_{ref} and x_{tar} to be aligned with the direction between the embeddings of y_{ref} and y_{tar} in the CLIP space. Secondly, they find fine-tuning the diffusion model is more effective than modifying the latent of the original image x_{ref} after diffusion inversion. Specifically, the original image x_{ref} is firstly converted to the latent using DDIM inversion. Then starting from the same latent, the diffusion model at the reverse path is fine-tuned, manipulating the generated image x_{tar} towards y_{tar}.

Additionally, Asyrp [70] discovered that changing the feature maps from the bottleneck layer of Unet could achieve desirable properties, such as homogeneity, linearity, robustness, and consistency across timesteps, to accommodate semantic image manipulation. To achieve fine-grained manipulation direction, ChatFace [81] utilizes a set of pre-defined text prompts and trains the manipulation directions with the supervision of CLIP direction loss.

Despite the advantages of using CLIP for image manipulation, the main limitation is that each fine-tuned model only works for one kind of editing instruction, i.e. manipulating the image aligned with y_{ref} towards the image aligned with y_{tgt}. Although these approaches demonstrate robustness in handling unseen domain images and benefit from highly optimized training schemes, the resulting manipulations may still not always meet desired application standard.

2.3.3 Representation Learning with Diffusion Models

Diffusion models often lack inherent semantic meaning in their latent variables, making them less suitable for representation learning, as emphasized by [82]. Representation learning involves the process of discovering meaningful higher-level concepts within data, such as identifying diverse topics in news articles, recognizing facial features in human photographs, or uncovering clusters of related molecules, all without explicit supervision.

To overcome this limitation and achieve a more meaningful latent code, DiffAE [17] introduces an additional image encoder that explores a semantically rich space, resulting in impressive capabilities for image inversion and reconstruction. Afterwards, PDAE [2] addresses the issue by training a gradient estimator functioned as classifier guidance. It utilizes the information from compact representations of input images to help reconstruction. This gradient estimator, denoted as $G(x_t, z, t)$, simulates $\nabla_{\mathbf{x}_t} \log p(z \mid \mathbf{x}_t)$. Additionally, InfoDiffusion [83] enhances the DiffAE [17] by incorporating a regularization term that helps increase the mutual information between the latent features and the produced samples, thereby improving the quality of the latent space.

Unlike many existing methods, our approach focuses on exploring disentangled representations with the additional encoder, which effectively captures interpretable and higher-level variances within the data.

Chapter 3

Methodology

We start by formally defining disentangled representations that we seek. Each sample $\mathbf{x} \in \mathcal{X}$ in the image space $\mathcal{X}$ is generated from a hidden attribute vector $\mathbf{z} \in \mathcal{Z}$ in the vector space $\mathcal{Z}$ through a mapping $\Phi : \mathcal{Z} \to \mathcal{X}$ (*i.e.*, different samples are generated by different attributes). We assume $\mathcal{Z}$ is decomposable. The disentanglement can be seen as decomposing the space $\mathcal{Z}$ into several subspaces $\mathcal{Z} = \mathcal{Z}_1 \times \ldots \times \mathcal{Z}_N$, where each $\mathcal{Z}_i$ is called a *modular* attribute, as its value can be locally intervened without affecting other modular attributes (*e.g.*, changing "expression" without affecting "gender" or "age"). A disentangled representation is the mapping $f : \mathcal{X} \to \mathcal{Z}$ that inverses Φ, *i.e.*, given each image $\mathbf{x}$, $f(\mathbf{x})$ recovers the modular attributes as a feature vector $\mathbf{z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$. The properties and common verification approaches are:

Linearity. A disentangled feature can be easily decoded using linear transformations, allowing for effective inference in downstream tasks. It is a common standard to validate through linear classification protocol.

Modularity. Latent dimensions should encode diverse and modular generative factors, such as object identities, positions, textures, and lighting conditions. It could be verified through image manipulation. Given an input image $\mathbf{x}$, we can intervene on the i-th modular attribute using an editing vector g_i, denoted as $g_i \cdot \mathbf{x} \in \mathcal{X}$ (for example, *increasing the smile* in $\mathbf{x}$). The process involves first mapping $\mathbf{x}$ to its corresponding latent representation $\mathbf{z}$ through the function f. Next, we apply the editing vector g_i to modify only the specific attribute $\mathbf{z}_i$ (e.g., $g_i =$

smile) while leaving other attributes unchanged. Finally, the edited latent representation $\mathbf{z}$ is fed into the function Φ for generating the modified output $\mathbf{x}$. This modular editing mechanism allows us to manipulate some individual attributes of an image while maintaining the remaining attributes or structures of the image.

In what follows, Section 3.1 reviews the key components of our approach: DDPM [67], DDIM inversion [84], classifier guidance [11] and SimSiam [18]. These components serve as the building blocks for our model's architecture. Section 3.2 describes the principle, algorithm, and design considerations of our proposed model. The illustration of the overall architecture of DiffSiam can be found in Figure 3.1.

3.1 Preliminary

3.1.1 Denoising Diffusion Probabilistic Models

Denoising Diffusion Probabilistic Model [67] (DM) is a latent variable model. It comprises a pre-defined forward encoding process, and a learnable reverse decoding process.

The forward process is set as a Markov chain that gradually moves each sample $\mathbf{x}_0$ from data distribution to the Gaussian distribution in the T-th time-step. It is common to pre-define constant variance schedule $\beta_1, \ldots, \beta_T$. They quantify how much noise is added at each time-step along the forward process. The formula is:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}). \tag{3.1}$$

In convenience, the distribution of $\mathbf{x}_t$ can be expressed with the initial sample $\mathbf{x}_0$ as:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}), \quad \text{where } \alpha_t := 1-\beta_t,\ \bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s. \tag{3.2}$$

The reverse process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is trained to approximate the posterior $q\left(\mathbf{x}_{t-1}|\mathbf{x}_t, u_\theta(\mathbf{x}_t, t)\right)$. The initial point of the reverse process is $p(\mathbf{x}_T) := \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$. Each $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is computed in two steps: 1) Reconstruct $\mathbf{x}_0$ from $\mathbf{x}_t$ with $u_\theta(\mathbf{x}_t, t)$, where u_θ is

a learnable U-Net [68]. 2) Compute $q\left(\mathbf{x}_{t-1}|\mathbf{x}_t, u_\theta(\mathbf{x}_t, t)\right)$, which has a closed-form solution given by $\mathcal{N}\left(\mathbf{x}_{t-1} \mid \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right), \tilde{\beta}_t \mathbf{I}\right)$:

$$\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right)=\frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0+\frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t} \mathbf{x}_t, \tilde{\beta}_t=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t \tag{3.3}$$

The training objective is equivalent to minimizing the reconstruction loss produced by the U-Net at any time-step t:

$$\mathcal{L}_{DM}=\mathop{\mathbb{E}}_{t, \mathbf{x}_0, \epsilon} \frac{\bar{\alpha}_t}{\left(1-\bar{\alpha}_t\right)^2}\|\mathbf{x}_0-u_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0+\sqrt{1-\bar{\alpha}_t} \epsilon, t)\|^2, \tag{3.4}$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a Normal Gaussian. For convenience, we formulate the U-Net to predict denoised image $\mathbf{x}_0$ instead of the noise term ϵ as in DDPM [67].

3.1.2 DDIM Inversion

The denoising diffusion implicit model (DDIM) [84] represents a modified version of DDPM that introduces a non-Markovian approach specifically in the training and sampling step. The benefits of relaxing the assumption of DDPM into a non-Markovian process are that 1) the sampling process becomes deterministic; 2) sampling is able to use trained diffusion models off-the-shelf. Specifically, one can generate $x_{t-\Delta t}$ from x_t via:

$$\frac{\boldsymbol{x}_{t-\Delta t}}{\sqrt{\alpha_{t-\Delta t}}}=\frac{\boldsymbol{x}_t}{\sqrt{\alpha_t}}+\left(\sqrt{\frac{1-\alpha_{t-\Delta t}}{\alpha_{t-\Delta t}}}-\sqrt{\frac{1-\alpha_t}{\alpha_t}}\right) \epsilon_\theta^{(t)}\left(\boldsymbol{x}_t\right) \tag{3.5}$$

It corresponds to an ordinary differential equation (ODE) after reparameterizing $(\sqrt{1-\alpha}/\sqrt{\alpha})$ with σ and $(\boldsymbol{x}/\sqrt{\alpha})$ with $\overline{\boldsymbol{x}}$:

$$\mathrm{d}\overline{\boldsymbol{x}}(t)=\epsilon_\theta^{(t)}\left(\frac{\overline{\boldsymbol{x}}(t)}{\sqrt{\sigma^2+1}}\right) \mathrm{d}\sigma(t) \tag{3.6}$$

This suggests that we can utilize the forward process to obtain a latent representation capable of accurately reconstructing the original input image, which we refer to as DDIM inversion. This property allows us to perform precise reconstruction,

leveraging the inherent capabilities of the DDIM framework. By leveraging DDIM inversion, we can achieve exact and faithful reconstruction of the input images.

3.1.3 Classifier Guidance

Classifier guidance is one of the techniques to add condition information into a pre-trained unconditional diffusion model without retraining. It commonly guides the reverse process with a gradient term of a trained classifier. Formally, during reverse process, we now sample $\mathbf{x}_{t-1}$ from:

$$\mathcal{N}\left(\mathbf{x}_{t-1} \mid \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right)+s \tilde{\beta}_t \nabla_{\mathbf{x}_t} \log p_\phi\left(y \mid \mathbf{x}_t, t\right), \tilde{\beta}_t \mathbf{I}\right) \tag{3.7}$$

For the deterministic sampling methods like DDIM, by using a score-based conditioning trick adapted from [10], one can formulate the conditional sampling as

$$\hat{u}_\theta(\mathbf{x}_t, t) := u_\theta(\mathbf{x}_t, t)-\sqrt{1-\bar{\alpha}_t} \nabla_{\mathbf{x}_t} \log p_\phi\left(y \mid \mathbf{x}_t\right) \tag{3.8}$$

3.1.4 SimSiam

SimSiam [18] network involves utilizing two randomly augmented views, denoted as $t_1(x_0)$ and $t_2(x_0)$, of the input image x_0 as inputs, where $t_1 \sim T_1, t_2 \sim T_2$ are two data augmentations. f_ω is the encoder network parameterizeed by ω, the p_γ is the predictor network parameterized by γ. SimSiam utilizes the siamese network [85] to process the two views. SimSiam's loss function is defined by:

$$\begin{aligned}
&\mathcal{L}_{\text{SimSiam}}\ (\omega, \gamma)= \\
&\mathbb{E}_{(\boldsymbol{x_0}, t_1, t_2)}\left[\frac{1}{2} \mathcal{D}(p_\gamma\left(f_\omega\left(t_1(\boldsymbol{x_0})\right)\right), f_\omega\left(t_2(\boldsymbol{x_0})\right))+\frac{1}{2} \mathcal{D}(p_\gamma\left(f_\omega\left(t_2(\boldsymbol{x_0})\right)\right), f_\omega\left(t_1(\boldsymbol{x_0})\right))\right]
\end{aligned} \tag{3.9}$$

$\mathcal{D}$ is the distance metrics formulated by:

$$\mathcal{D}\left(p_1, z_2\right)=-\frac{p_1}{\left\|p_1\right\|_2} \cdot \frac{z_2}{\left\|z_2\right\|_2} \tag{3.10}$$

As a baseline approach, we choose SimSiam due to its simplicity and competitive performance with fewer computational resources.

3.2 DiffSiam

We observe that attributes with finer granularity, which result in fewer visible changes at the pixel level, tend to be lost at earlier time-steps compared to attributes with coarser granularity. This leads to two interesting groundings: 1) Attribute loss is linked with the error of reconstructing $\mathbf{x}_0$ from the noisy $\mathbf{x}_t$. 2) Increasing time-step t in the forward diffusion process causes an expanding cumulative loss of attributes, with fine-grained attributes lost at a smaller t and coarse-grained ones lost at a larger t.

These insights provide a strong inductive bias for unsupervised learning frameworks:

1) Firstly, in models dealing with reconstruction (like autoencoders [86, 87]), a common challenge is the loss of certain attributes or features in the reconstructed output due to reconstruction errors. The inductive bias in this context refers to our model's inherent tendency or strategy to compensate for these errors. Specifically, our encoder is trained in a way that it actively tries to compensate the reconstruction error of the diffusion model, so that the feature vector inferred with the encoder is able to capture semantic attributes of the input image.

2) Secondly, as illustrated in Figure 1.1, we observe that attributes with finer granularity, which result in fewer visible changes at the pixel level, tend to be lost at earlier time-steps compared to attributes with coarser granularity. Based on this observation, our encoder is designed to use different subset of feature vector to learn the attributes in different granularity. The inductive bias in this context refers to our model's inherent tendency to capture different semantic attributes with separate dimensions of the feature vector.

These principles guide the proposed approach. As illustrated in Figure 3.1, our model includes three parts: (1) a trainable encoder f_ω (trained to become a disentangled representation) that maps each image (without noise) to a d-dimensional

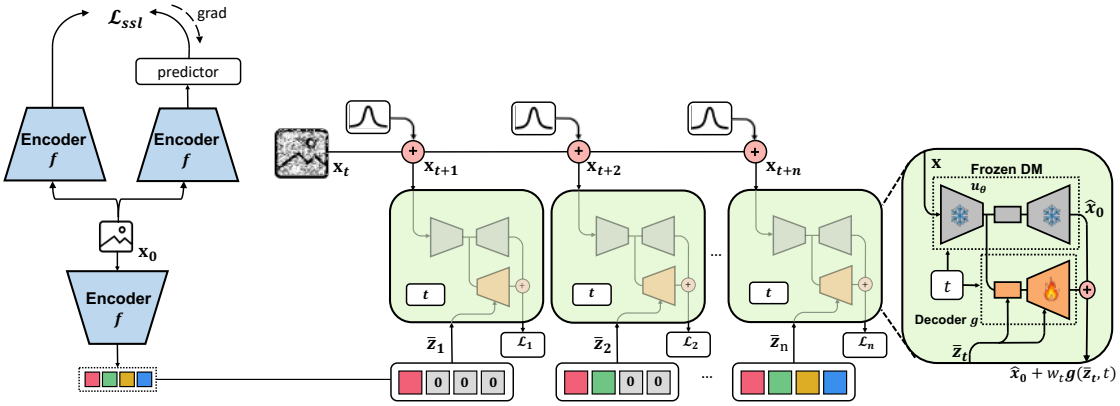


FIGURE 3.1: Architecture of our DiffSiam. On the left, we train a SimSiam network and encode the image x_0 to the representation with n segments. On the middle, we break down the learning with Eq. (3.11) at each time -step. On the right, we show the interal network design. We follow PDAE [2] for the decoder design. $\hat{\mathbf{x}}_0$ denotes the reconstructed $\mathbf{x}_0$ by the pre-trained DM.

feature $\mathbf{z} \in \mathbb{R}^d$, (2) a frozen DM U-Net u_θ pre-trained using Eq. (3.4), and (3) a trainable decoder g_γ that maps $\mathbf{z}$ back to image space given t.

During training, the encoder does two things. The first thing is to follow the conventional SimSiam training to learn representations that makes closer the two augmentations of the input image x_0. The training objective of SimSiam is given in Section 3.1.4. The other thing is to prepare the latents for disentangled representation learning in the diffusion model. Given $\mathbf{z} = f_\omega(\mathbf{x}_0)$, we evenly partition it into n disjoint subsets $\{\mathbf{z}_i\}_{i=1}^n$ along the feature dimension. We construct a new representation $\bar{\mathbf{z}}_k = [\mathbf{z}_1, \ldots, \mathbf{z}_k, \mathbf{0}, \ldots, \mathbf{0}]$ $(k \in [1, n])$. The representation is only effective within $\{\mathbf{z}_i\}_{i=1}^k$ by masking $\mathbf{z}_{k+1}, \ldots, \mathbf{z}_n$ as $\mathbf{0}$. Masking is used for maintaining the independence between disjoint feature subsets. For example, when we train the subset $\mathbf{z}_k$, we only want $\mathbf{z}_k$ to learn the lost attribute at the current time-step, so we need to mask other subsets $\mathbf{z}_{k+1}, \ldots, \mathbf{z}_n$. Subsets $\mathbf{z}_1, \ldots, \mathbf{z}_{k-1}$, although not masked due to their essential fine-grained information helpful for reconstruction, have their gradients detached to maintain the independence between them and $\mathbf{z}_k$. This optimization strategy is discussed in Section 4.3.2.

For all the datasets and evaluation protocols, we use ResnNet-10 backbone as the encoder. Regarding the number of subsets n, a small value might miss capturing all the underlying attributes, while a larger value of n could lead to each $\mathbf{z}_i$ having a very small dimension, potentially lacking the capacity needed to capture

any specific hidden attributes. In later experiments, we empirically partition the representation $\mathbf{z}$ into 64 segments, *i.e.* $n = 64$.

Moreover, at each time-step t, we do additional k forward steps to the current noisy image $\mathbf{x_t}$. Given that the noise scheduler gradually approximately converts the input image into a pure Gaussian noise at timesteps $T = 1000$, increasing the number of additional forward steps could introduce more diverse attribute loss. As a result, we obtain the noisy image x_{t+k} that has additional attributes being lost.

Then the diffusion objective requires $\bar{\mathbf{z}}_k$ to compensate for the reconstruction error made by u_θ. In other words, the diffusion model tries to reconstruct the ground truth image $\mathbf{x}_0$ with $\hat{\mathbf{x}}_0$ at each time-step t. We employ the decoder g parameterized by ϕ to decode $\bar{\mathbf{z}}_k$ back to image space as $g_\phi(\bar{\mathbf{z}}_k, t)$. The decoder design follows in PDAE [2]. Specifically, the decoder trains the middle blocks, up-sample layers and output blocks of the U-Net, and gets the input from the frozen U-Net encoder. From the perspective of classifier guidance, the output of the decoder is to simulate the gradient: $\nabla_{\mathbf{x}_t} \log p_\phi\left(\bar{\mathbf{z}}_k \mid \mathbf{x}_t, t\right)$.

The training objective requires $\bar{\mathbf{z}}_k$ to compensate for the information of attribute loss by u_θ at each time step, given by:

$$\mathcal{L}_{\text{diffusion}} = \mathop{\mathbb{E}}_{t,\mathbf{x}_0,\epsilon,k} \underbrace{\lambda_t \|\mathbf{x}_0 - \left(u_\theta(\sqrt{\bar{\alpha}_{t+k}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t+k}}\epsilon, t) + w_t g_\phi(\bar{\mathbf{z}}_k, t)\right)\|^2}_{\mathcal{L}_k}, \quad (3.11)$$

where λ_t, w_t are time-step weight and compensate strength, which are fixed values computed from the variance schedule $\{\beta_t\}_{t=1}^T$, and we follow the settings in PDAE [2]. k represents the count of additional forward steps introduced.

To see the effect of Eq. (3.11), we consider the following induction. We assume that attributes of an image can be decomposed into N modular attribute $\mathcal{Z}_1, \ldots, \mathcal{Z}_N$ (e.g., expression, age, gender, etc.). When $k = 1$, only the subset $\mathbf{z}_1$ is used to compensate for reconstruction error, hence $\mathbf{z}_1$ effectively captures a (fine-grained) modular attribute $\mathcal{Z}_i$ $(i \in [1, N])$. Then when $k = 2$, the cumulatively lost attributes correspond to $\mathcal{Z}_i$ and $\mathcal{Z}_j$ $(j \in [1, N], j \neq i)$, and because $\mathbf{z}_1$ already captures the attribute $\mathcal{Z}_i$, $\mathbf{z}_2$ is encouraged to further learn the attribute $\mathcal{Z}_j$ to minimize the reconstruction error. Eventually with a large enough k, $\mathbf{x}_{t+k}$ gets

closer to a pure noise and almost all attributes are lost. By the above induction, $\{\mathbf{z}_i\}_{i=1}^n$ captures all the hidden attributes as modular features. In contrast, the most related work PDAE differs from Eq. (3.11) by replacing $g(\bar{\mathbf{z}}_t)$ with $g(\bar{\mathbf{z}})$. This means that PDAE uses the full-dimensional feature $\mathbf{z}$ to compensate for the reconstruction error at all time-steps, which fails to leverage the inductive bias. We show in Chapter 4 that this simple change leads to drastic differences in both counterfactual generation and manipulation tasks.

Finally, the total loss can be written as, given Eq. (3.11) and Eq. (3.9):

$$\mathcal{L} = \mathcal{L}_{\text{diffusion}} + \mathcal{L}_{\text{SimSiam}} \tag{3.12}$$

Please refer to Algorithm 1 for the training algorithm.

Algorithm 1: DiffSiam Training

Input : Training data distribution $q(\mathbf{x}_0)$, pre-trained DM u_θ

Output: Trained encoder f_ω, decoder g_ϕ, predictor network p_γ

Randomly initialize ω, ϕ, γ;

while *not converged* **do**

- $\mathbf{x}_0 \sim q(\mathbf{x}_0)$;
- $\mathbf{z} = f(\mathbf{x}_0)$;
- Partition $\mathbf{z}$ into $\{\mathbf{z}_i\}_{i=1}^n$;
- $t \sim \text{Uniform}(1, T-n)$;
- $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
- $k \sim \text{Uniform}(1, n)$;
- $\mathbf{x}_{t+k} = \sqrt{\bar{\alpha}_{t+k}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t+k}}\boldsymbol{\epsilon}$;
- $\bar{\mathbf{z}}_k = [\mathbf{z}_1, \ldots, \mathbf{z}_k, \mathbf{0}, \ldots, \mathbf{0}]$;
- $\mathcal{L} = \mathcal{L}_{\text{diffusion}} + \mathcal{L}_{\text{SimSiam}}$;
- Update $f_\omega, g_\phi, p_\gamma$ by minimizing $\mathcal{L}$;

return $f_\omega, g_\phi, p_\gamma$

We highlight some design considerations with ablations in Section 4.3. We choose an accumulative set of the latent dimension $\mathbf{z}$ to compensate for the lost attribute information as time-step increases. We name it the *Hierarchical* DiffSiam. Specifically, if we are going to do additional k forward steps to the current noisy image, then we guide the reconstructed image with $\bar{\mathbf{z}}_k = \sum_{i=1}^k \mathbf{z}_k$, i.e. the representation from the first k segments of $\mathbf{z}$. We observe that the choice of accumulative

representation learning improves the performance of classification tasks, especially few-shot learning. It aligns with the intuition that late time-step introduces more attribute loss so that it should be compensated with more dimensions of $\mathbf{z}$. By contrast, employing a single segment $\mathbf{z}_k$ to compensate the new attribute loss, named as *Sequential* DiffSiam, falls short of disentangling features. Along with the choice of *Hierarchial* DiffSiam, we experiment with detaching the gradient of $\mathbf{z}_1, \ldots, \mathbf{z}_{k-1}$ to form $\bar{\mathbf{z}}_k$, *i.e.*, training only $\mathbf{z}_k$ at time-step t while still using the information of the first k segments. We find that it leads to improved disentanglement quality at the cost of slower convergence.

Chapter 4

Experiments

4.1 Classification

Datasets and Settings

We use the pre-trained DDPM [67] on Cifar-10 [88] and train the encoder, decoder on Cifar-10 with a learning rate of 0.05. Our Cifar-10 model was trained for 400k iterations with a batch size of 128, taking about 1.5 days on an NVIDIA RTX A6000.

We leverage the pre-trained Denoising Diffusion Probabilistic Model (DDPM) [67] initially trained on the Cifar-10 dataset [88]. Subsequently, we further train the encoder and decoder on the Cifar-10 dataset using a learning rate of 0.05. It takes our Cifar-10 model about 400,000 iterations with a batch size of 128. Training takes about 1.5 days on one NVIDIA RTX A6000.

After obtaining the trained encoder network, we add a linear layer atop the encoder. We further fine-tune the linear layer for linear classification, many-shot recognition and few-shot learning.

We report the Cifar-10 [88] linear evaluation performance. Also, we examined the transferability of the representation through many-shot and few-shot classification following [89, 90], and present the classification accuracies on Cars [91], Cifar-100 [88], DTD [92], Food [93], SUN [94]. Additionally, we report the average per-class accuracies on Aircraft [95], Caltech [96], Flowers [97], Pets [98]. These datasets

Method	CIFAR10	Aircraft	Caltech	Cars	CIFAR100	DTD	Flowers	Food	Pets	SUN	Avg.
SimSiam	**90.980**	21.795	54.626	11.479	50.480	34.096	46.285	24.840	39.120	19.632	39.333
DiffSiam (Seq.)	90.710	**27.307**	57.467	**15.769**	56.500	33.670	**55.433**	28.824	**43.504**	23.280	43.246
DiffSiam (Hier.)	90.880	27.097	**60.246**	15.271	**57.730**	**35.479**	54.822	**29.374**	42.545	**24.146**	**43.759**

Table 4.1: Accuracy (%) of many-shot linear classification results using the encoder learned on Cifar-10. We also include the Cifar-10 evaluation performance. The best performance score is shown in bold.

vary in the size of training data (2,000-75,000 images) and category count (10-397 classes) .

4.1.1 Many-shot Recognition

Results We present our results and conduct a comparison with the SimSiam [18] baseline method in Table 4.1. We see that: (a) DiffSiam, including Hierarchial and Sequential architecture, brings consistent performance boosts on various datasets that are slightly shifted from the domain of the dataset that the encoder is pre-trained on. For example, It results in performance increases of 5.62% on Caltech and 7.25% on Cifar100. This validates that our algorithm is better at capturing higher-level and domain-agnostic features that are transferable across different datasets. (b) Our approach achieves slightly lower results on the original pre-trained dataset, CIFAR-10. It can be explained that CIFAR-10 is balanced and structured with distinct class boundaries. On the other hand, the other datasets we evaluated, like Aircraft, Caltech, Cars, have varying levels of complexity, class imbalance, or intra-class variability, where our approach's disentanglement makes a bigger difference. The average score provides empirical validation that our approach enhances disentanglement in comparison to the conventional SSL method.

4.1.2 Few-shot Learning

A few-shot K-way C-shot image classification setting contains a collection of tasks, often referred to as episodes. A single K-way C-shot episode is illustrated as: suppose the entire training set contains N the total number of classes. In the construction of each episode, we randomly choose K classes from N. Training samples from the K classes are randomly chosen to form a support set and a query set: (a) the support set contains K samples for each class; (b) the query set is selected from the remaining instances within each class. We conducted a

	Method	CIFAR10	Aircraft	Caltech	Cars	CIFAR100	DTD	Food	Pets	SUN
1-shot	SimSiam	37.44±.34	31.72±.40	50.14±.51	27.95±.30	39.54±.42	31.95±.34	29.25±.30	35.45±.36	38.59±.39
	DiffSiam (Seq.)	40.98±.34	33.52±.40	56.47±.52	28.79±.31	45.09±.43	34.25±.37	31.43±.32	37.70±.37	43.56±.41
	DiffSiam (Hier.)	**43.20**±.35	**33.89**±.42	**57.25**±.53	**29.10**±.31	**46.22**±.45	**34.59**±.36	**31.59**±.32	**38.13**±.38	**44.29**±.42
5-shot	SimSiam	70.48±.33	40.69±.49	74.55±.42	36.98±.38	64.58±.42	47.46±.39	43.08±.36	53.84±.40	62.81±.40
	DiffSiam (Seq.)	70.66±.32	44.30±.50	78.27±.38	38.90±.39	68.15±.40	**49.82**±.40	44.99±.37	55.34±.39	66.01±.39
	DiffSiam (Hier.)	**74.05**±.32	**44.40**±.51	**79.11**±.39	**39.25**±.38	**69.34**±.41	49.76±.41	**45.22**±.37	**55.90**±.40	**67.28**±.39
10-shot	SimSiam	81.74±.27	44.03±.50	80.48±.35	41.54±.40	71.38±.39	53.25±.39	48.51±.36	60.21±.38	69.01±.37
	DiffSiam (Seq.)	80.38±.27	48.09±.50	83.67±.32	43.87±.40	74.46±.38	**55.46**±.39	50.82±.37	61.89±.38	71.68±.36
	DiffSiam (Hier.)	**82.76**±.26	**48.32**±.51	**84.28**±.32	**44.31**±.40	**75.44**±.37	55.28±.39	**50.89**±.36	**62.28**±.38	**73.02**±.36

Table 4.2: 5-way few-shot evaluation accuracies (%) using the Cifar-10 trained encoder. Mean accuracy and standard deviation are reported from 2,000 episodes. Best performance per dataset is highlighted in bold.

random sampling of 2,000 episodes from the test split of each dataset. Each episode comprised n classes, with k samples for training and 15 samples for queries per class. The linear classifier was subsequently fine-tuned with a frozen backbone across the training samples over the course of 100 epochs.

In few-shot learning, a setup called a K-way C-shot image classification involves tasks, often termed episodes. Suppose we have N classes in the entire training dataset. For each episode, we randomly pick K classes from these N. We then select C samples from each of these K classes to form a support set, and also pick a certain number of samples from the remaining samples of the same class for a query set. In our experiments, we set up a 5-way few-shot evaluation with shot count $C = 1, 5, 10$. 2,000 episodes are sampled from the test dataset. Each episode has 15 samples for queries per class. We then fine-tune the linear classifier using the frozen backbone across the training samples for 100 epochs. Across methods and datasets, the classifier is trained over 100 epochs with batch size 4. We utilize the SGD optimizer with the initial learning rate as 0.01 and the parameter of weight decay as 0.001. We evaluate the linear classifier on 15 query images for each episode.

Results In Table 4.2, our DiffSiam method demonstrates substantial performance improvement across the 5-way few-shot learning. This is because DiffSiam can disentangle semantics more effectively than SSL approaches, which promotes the generalizability of representations across diverse data distributions. This observation aligns with the research [99], which emphasizes the benefits of disentangled representations, particularly in scenarios involving limited data (few-shot). It further highlights the significance of disentanglement in enhancing performance across downstream tasks.

4.2 Generative Tasks

Datasets and Settings.

We adopted 3 training datasets: 1) Celebrity Faces Attributes (CelebA) [1] has 40 attribute labels per image, including various facial features and characteristics such as smiling, wearing glasses, gender, etc. 2) Flickr-Faces-HQ (FFHQ) [55] comprises 70,000 human face images. 3) Bedroom is part of LSUN [56] containing about 3 million images, depicting bedrooms in a wide variety. We used the pre-trained DM trained on FFHQ, CelebA, and Bedroom provided by.

Our model training utilizes PDAE's [2] pre-trained DM. The model is trained for 290k iterations on CelebA, 500k iterations on FFHQ, and 540k iterations on Bedroom, with a batch size of 128 consistent with PDAE. CelebA images were resized to 64×64 while FFHQ and Bedroom images were resized to 128×128. We employ the Adam optimizer with a learning rate of $1e^{-4}$. To train our DiffAE model, we employed four NVIDIA A100 40GB with batch size 32 per device. Training on the FFHQ and Bedroom datasets generally requires approximately 5 days, while training on CelebA takes around 1 day.

4.2.1 Counterfactual Generation

Our objective is to create counterfactual images by interpolating a subset $\mathcal{S}$ of the n modular attributes. For a given image pair $\mathbf{x}_0$ and $\mathbf{x}'_0$, we utilize DDIM inversion [84] through the pre-trained DM to derive the DDIM-inversed counterparts $\mathbf{x}_T$ and $\mathbf{x}'_T$. The modular attributes $\mathbf{z} = \{\mathbf{z}_i\}_{i=1}^n, \mathbf{z}' = \{\mathbf{z}'_i\}_{i=1}^n$ are obtained by encoding $\mathbf{x}_0$ and $\mathbf{x}'_0$ using the function f.

To generate counterfactuals for $\mathbf{x}_0$, we initiate the process from the noisy sample $\text{Slerp}(\mathbf{x}_\text{T}, \mathbf{x}'_\text{T}; \lambda)$—employing spherical linear interpolation [100] of $\mathbf{x}_T$ and $\mathbf{x}'_T$ with scale λ. Subsequently, we execute DDIM sampling, guided by $g(\text{lerp}(\mathbf{z}, \mathbf{z}', \mathcal{S}; \lambda), t)$, where only linear interpolation (lerp) is performed with scale λ on the subset $\mathcal{S}$ of $\mathbf{z}$, leaving the values of other modular attributes unchanged. The process for generating counterfactuals for $\mathbf{x}'_0$ follows a similar procedure. Please refer to Algorithm 2.

Algorithm 2: Counterfactual generation from $\mathbf{x}_0$ to $\mathbf{x}'_0$ on subset $\mathcal{S}$ with scale λ

Input : $\mathbf{x}_0$, $\mathbf{x}'_0$, subset $\mathcal{S} \subset \{1, \ldots, k\}$, scale λ, pre-trained u_θ, trained encoder f_ω and decoder g_ϕ, sampling sequence $\{t_i\}_{i=1}^M$ where $t_1 = 0$ and $t_M = T$

Output: A counterfactual image for $\mathbf{x}_0$

Compute $\mathbf{x}_T, \mathbf{x}'_T$ for $\mathbf{x}_0, \mathbf{x}'_0$ with DDIM inversion, respectively;

$\mathbf{z} = f_\omega(\mathbf{x}_0), \mathbf{z}' = f_\omega(\mathbf{x}'_0)$;

Partition $\mathbf{z}$ into $\{\mathbf{z}_i\}_{i=1}^n$, $\mathbf{z}'$ into $\{\mathbf{z}'_i\}_{i=1}^n$;

$\mathbf{z}_\mathcal{S} \leftarrow \text{lerp}(\mathbf{z}, \mathbf{z}', \mathcal{S}; \lambda)$, *i.e.*, perform linear interpolation on all $\mathbf{z}_i, i \in \mathcal{S}$;

$\mathbf{x}_T \leftarrow \text{slerp}(\mathbf{x}_T, \mathbf{x}'_T; \lambda)$;

for $i = M, \ldots, 2$ **do**

$\hat{\mathbf{x}}_0 = u_\theta(\mathbf{x}_{t_i}, t_i) + w_t g_\phi(\mathbf{z}_\mathcal{S}, t_i)$;

$\mathbf{x}_{t_{i-1}} \leftarrow \sqrt{\bar{\alpha}_{t_{i-1}}}\hat{\mathbf{x}}_0 + \frac{\sqrt{1-\bar{\alpha}_{t_{i-1}}}(\mathbf{x}_{t_i} - \sqrt{\bar{\alpha}_{t_i}}\hat{\mathbf{x}}_0)}{\sqrt{1-\bar{\alpha}_{t_i}}}$;

return $\mathbf{x}_0$

Results on FFHQ. In Figure 4.1, we show counterfactual generations on FFHQ with different feature subsets. For each pair associated with a specific subset $\mathcal{S}$, the image at the top left and the bottom right corresponds to $\mathbf{x}_0$ and $\mathbf{x}'_0$ respectively. Starting from $\mathbf{x}_0$, counterfactual generation involves interpolation with $\mathbf{x}'_0$ at three different scales (1/3, 2/3, and 1), with the results arranged from left to right. The same process is applied to $\mathbf{x}'_0$, and the outcomes are presented from right to left. Note that a scale of 1 signifies the replacement of feature values associated with the subset $\mathcal{S}$ using those from the other sample. Empirically we found that the loss is low on the late feature subset due to the design of time-step weight of DDPM. Hence more feature subsets are grouped together to make more visible edits. Note that the scale of 1 means the feature values corresponding to $\mathcal{S}$ is replaced with those of the other sample.

We observe that interpolating subset 0-2 roughly corresponds to modifying "expression" attribute (*e.g.*, center pair). Interpolating 0-4 and 0-6 additionally modifies "mouth" attribute and "eyes" attribute, respectively. This shows that the learned feature subspace $\mathcal{Z}_1, \mathcal{Z}_2, \mathcal{Z}_3$ corresponds to the three attributes, and can be combined by Cartesian product, hence verifying our disentanglement quality. Subsets 6-10, 16-24, 24-32, 32-64 correspond to gradually coarser-grained attributes, "face shape", "face direction", "hairstyle", "decorations on the head", respectively. This

FIGURE 4.1: DiffSiam's counterfactual generation results on FFHQ. For each pair of images being interpolated, original images are positioned at the two corners, and then interpolated within a designated subset of $\{\mathbf{z}_i\}_{i=1}^{k}$ to gradually transition towards its counterpart with four scales (1/4, 2/4, 3/4 and 1). Each row is the result of interpolating on a subset, with the subset range displayed on the left.

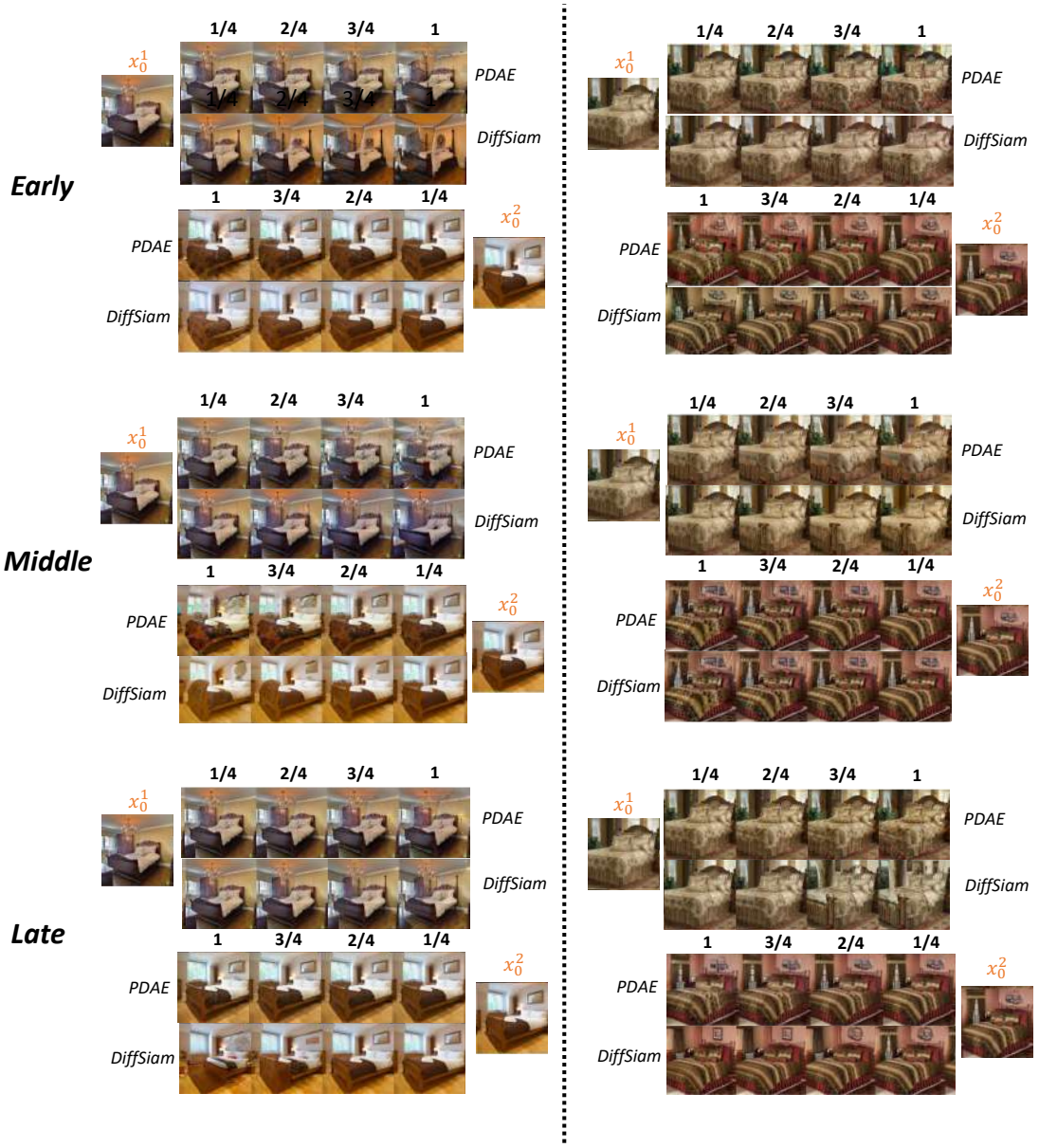


FIGURE 4.2: Comparisions on PDAE (top row) and DiffSiam (bottom row)'s counterfactual generation results on Bedroom using 3 feature subsets. For each image pair in each subset, the original images, $\mathbf{x}_0^1, \mathbf{x}_0^2$ are positioned at the upper-left and lower-right corners, and interpolated towards its counterpart in four scales (1/4, 2/4, 3/4 and 1) respectively.

strongly validates that our DiffSiam learns modular feature (*i.e.*, coarse-grained attributes are lost later and captured by $\mathbf{z}_i$ with a large i).

Bedroom Results We compare the counterfactual generation results between PDAE and DiffSiam on three subsets in Figure 4.2. "Early", "Middle", "Late" correspond to the subset 0-8, 8-16, 24-28 respectively. We observe that each subset

FIGURE 4.3: Comparsions of PDAE (top row) and DiffSiam(bottom row)'s interpolation results on FFHQ using all dimensions of the feature $\mathbf{z}$.

controls some meaningful attribute in the image. Take the right column for example, in the early subset, the overall color scheme is interpolated. Interpolating the middle subset changes the shape of the end of the bed. x_0^1 takes more red from x_0^2 with the increasing of the scale. On the other hand, x_0^2 gets more brown color when its scale increases. Interpolating the middle subset (8-16) changes the texture of the quilt. Interpolating the late subset changes the background. x_0^1's background changes from the glass window to the wall that x_0^2 has. At the same time, x_0^2 changes the photo frame on the wall. Although PDAE also changes the background in the late subset, it changes the texture in the quilt at the same time.

Full-dim Interpolation Results on FFHQ. We run the standard full-dim interpolation experiments (*i.e.*, interpolating the entire $\mathbf{z}$ instead of a subset) with results in Figure 4.3. We highlight that our DiffSiam outperforms PDAE, *e.g.*, more meaningful background changes, less artifact and more convincing intermediate results. This advantage can be attributed to the enhanced disentanglement and preservation capabilities of DiffSiam's representation, which contribute to generating higher-quality and more contextually consistent images during the interpolation process.

Full-dim Interpolation Results on Bedroom. In Figure 4.4, we show the interpolation results on Bedroom, where our DiffSiam (bottom row) outperforms PDAE (top row) with less artifact. Note that Bedroom is especially challenging for interpolation due to the large variations between images. Larger models and

FIGURE 4.4: Comparsions of PDAE (top row) and DiffSiam (bottom row)'s interpolation results on Bedroom using all dimensions of the feature $\mathbf{z}$.

prolonged training can benefit the interpolation fidelity. We leave it as future work as improving generation fidelity is not the focus of our work.

4.2.2 Attribute Manipulation

In modular manipulation, we use the attribute labels to train a linear classifier that predicts a specific attribute. On CelebA [1], we use its 40 attribute labels for training. On Bedroom [56], there are no ground-truth attribute labels. We use pseudo-labels produced by an off-the-shelf attribute predictor [101] to train the attribute classifier. In particular, we adopt probabilistic masking (ProbMask) [102], a network sparsification method to constrain the classifier such that its weight has only d' non-zero dimensions, where $d' < d$ (*e.g.*, $d' = 16$ or 32 and $d = 512$). This design is to test the modularity of the feature—a specific attribute (*e.g.*, "Young") should be captured by the combination of a few modular attributes $\mathbf{z}_i$, but not all. With the trained classifier for an attribute, to manipulate the attribute with scale λ on a sample $\mathbf{x}_0$, we first obtain its feature $\mathbf{z} = f_\omega(\mathbf{x}_0)$, then push $\mathbf{z}$ along the normal vector of the decision boundary with certain scale λ, resulting in manipulated code $\mathbf{z}'$, and finally encode $\mathbf{x}_T$ back to the manipulated image by the guidance of $g_\phi(\mathbf{z}', t)$. The process is summarized in Algorithm 3.

Algorithm 3: Modular manipulation on $\mathbf{x}_0$ with a trained ProbMask classifier and scale λ

Input : Original $\mathbf{x}_0$, manipulation scale λ, trained ProbMask classifier with weight parameter $\mathbf{w} \in \mathbb{R}^d$, pre-trained DM u_θ, trained encoder f_ω and decoder g_ϕ, standard deviation $\boldsymbol{\sigma}$ of $\mathbf{z}$ in the entire training dataset, sampling sequence $\{t_i\}_{i=1}^M$ where $t_1 = 0$ and $t_M = T$

Output: A counterfactual image for $\mathbf{x}_0$

Compute $\mathbf{x}_T$ for $\mathbf{x}_0$ with DDIM inversion;

$\mathbf{z} = f_\omega(\mathbf{x}_0)$;

$\mathbf{z}' = \mathbf{z} + \lambda \frac{\boldsymbol{\sigma} \cdot \mathbf{w}}{\|\mathbf{w}\|}$;

for $i = M, \ldots, 2$ **do**

$\hat{\mathbf{x}}_0 = u_\theta(\mathbf{x}_{t_i}, t_i) + w_t g_\phi(\mathbf{z}', t_i)$;

$\mathbf{x}_{t_{i-1}} \leftarrow \sqrt{\bar{\alpha}_{t_{i-1}}}\hat{\mathbf{x}}_0 + \frac{\sqrt{1-\bar{\alpha}_{t_{i-1}}}(\mathbf{x}_{t_i} - \sqrt{\bar{\alpha}_{t_i}}\hat{\mathbf{x}}_0)}{\sqrt{1-\bar{\alpha}_{t_i}}}$;

return $\mathbf{x}_0$

CelebA Results. Results are shown in Figure 4.5, where our DiffSiam makes more meaningful edits with fewer artifacts compared to PDAE. In particular, we use more ProbMask dimension $d' = 128$ for attribute "Wearing Hat", as it corresponds to larger edits that involve multiple modular attributes (*e.g.*, shadows and forehead).

Bedroom Results We show the manipulation results on Bedroom in Figure 4.6 for four attributes: "indoor lighting", "rusty", "cluttered space" and "carpet". Pushing the latent code along the normal direction is able to change the specific attribute in the image. For example, the room becomes much brighter and warm when the scale increases in the positive direction, while it becomes colder and dimmer in the negative direction. Note that here we only manipulate 32 dimensions of the latent code, much fewer than the latent code $\mathbf{z}$ has, *i.e.*, 512. It shows that our mode learns a disentangled and compact feature. In contrast, the manipulated images by PDAE often contain artifacts and have less meaningful edits, which further validates that PDAE entangles all the attributes.

Full-dim Manipulation Results on CelebA. We run the standard manipulation experiments (*i.e.*, without ProbMask) with results in Figure 4.7. Our results (bottom row) are on par or better than PDAE (top row), *e.g.*, generating hat with

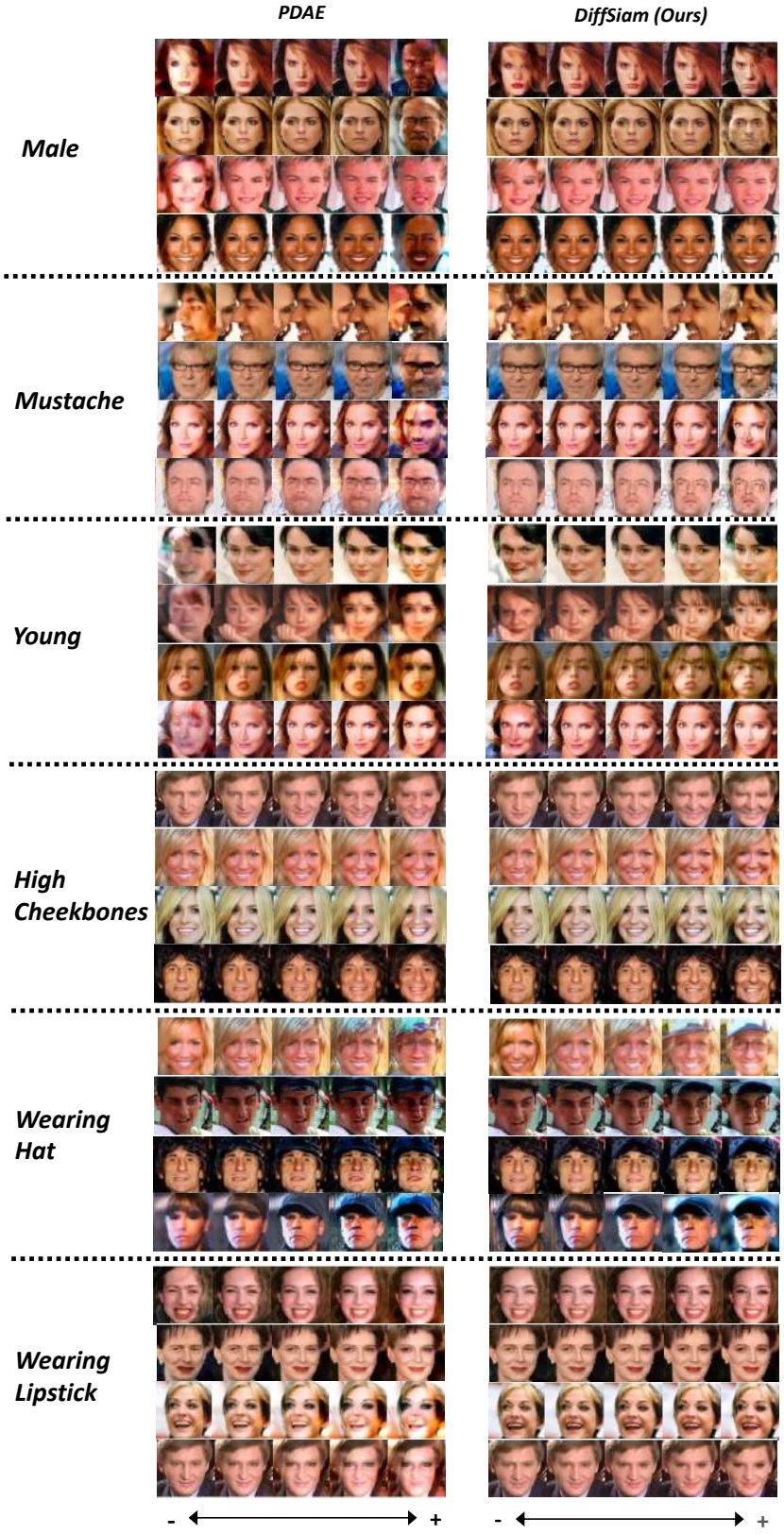


FIGURE 4.5: Comparsions of PDAE and DiffSiam's modular manipulation results on CelebA. The center image in each group of 5 images corresponds to the original image $\mathbf{x}_0$. We use $d' = 128$ as the dimension of ProbMask on "Wearing Hat" attribute and $d' = 16$ for the rest.

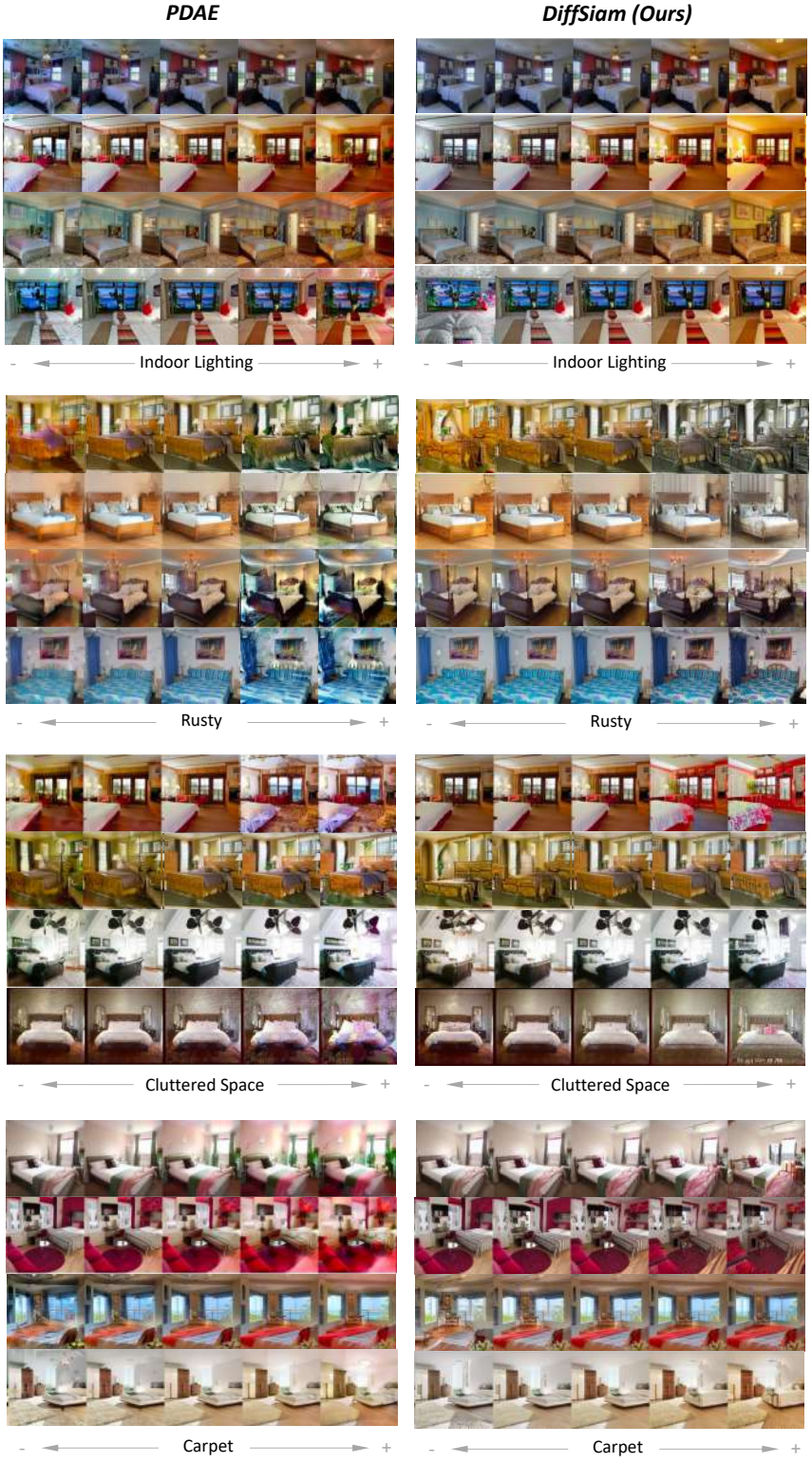


FIGURE 4.6: Comparsions of PDAE (left column) and DiffSiam (right column)'s modular manipulation results on Bedroom. For both methods, we use ProbMask to constrain the classifier weight such that it has $d' = 32$ non-zero dimensions.

FIGURE 4.7: Comparsions of PDAE (top row) and DiffSiam (bottom row)'s manipulation results on CelebA using all dimensions of the feature $\mathbf{z}$.

less artifact. This experiment is to demonstrate that DiffSiam still has the conventional manipulation capability, and the aforementioned modular manipulation experiments are more suitable to highlight our disentanglement quality.

4.3 Ablation Study

4.3.1 Hierarchial vs Sequential strategy

The illustration of two strategies is shown in Figure 4.8. When extending the current noisy image by an additional k forward steps, Hierarchial DiffSiam guides the reconstructed image with accumulated latent representation $\bar{\mathbf{z}}_k = \sum_{i=1}^{k} \mathbf{z}_k$, while Sequential DiffSiam only relies on the information from the current latent state $\mathbf{z}_k$. We experiment with Hierarchial and Sequential architecture in classification tasks. Upon revisiting the few-shot classification results in Tables 4.1 and 4.2, the Hierarchical DiffSiam consistently outperforms datasets featuring fine-grained categories, as seen in the cases of Caltech and SUN. These datasets require telling apart visually similar categories for accurate classification. This emphasizes how

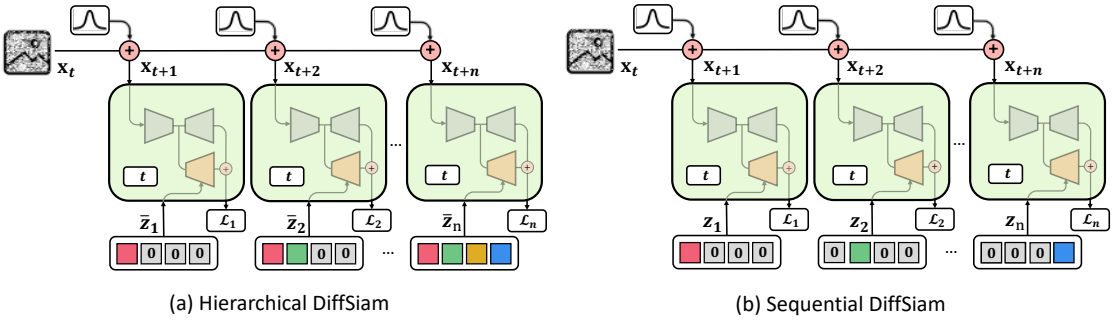


FIGURE 4.8: Comparsion between Hierarchial DiffSiam and Sequential DiffSiam. At each step $t + k$, Hierarchial one (a) guides the reconstructed image with accumulated latent representation $\bar{\mathbf{z}}_k = \sum_{i=1}^{k} \mathbf{z}_k$, while Sequential DiffSiam (b) only relies on the information from $\mathbf{z}_k$.

the hierarchical disentanglement strategy is helpful in uncovering important semantic features that stand out from each other. Conversely, Sequential DiffSiam struggles to utilize the semantic information within the latent space.

4.3.2 Optimization Strategy

Concerning the optimization strategy of Hierarchical DiffSiam, we experimented with detaching the gradients of $\mathbf{z}_1, \ldots, \mathbf{z}_{k-1}$ while guiding the reconstructed image with accumulated latents $\bar{\mathbf{z}}_k$. Although this approach led to improved disentanglement quality in the counterfactual generation task, it also resulted in slower convergence due to each $\mathbf{z}_i$ being trained only $\frac{1}{k}$ iterations. Figure 4.9 compares the loss in Eq. 3.11 by Hierarchical DiffSiam and Hierarchical DiffSiam-Detach (*i.e.*, detaching the gradients of $\mathbf{z}_1, \ldots, \mathbf{z}_{k-1}$) throughout training. The loss reduction of DiffSiam-Detach is much slower as only $\frac{1}{k}$ of the feature is trained at each iteration.

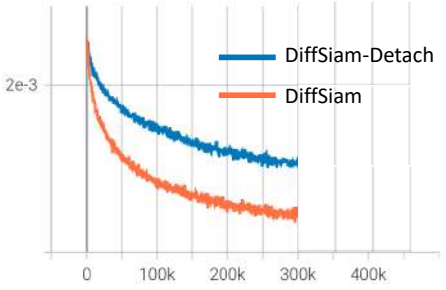


FIGURE 4.9: Training loss of DiffSiam and DiffSiam with detach optimiaztion strategy.

As future work, we will explore improved network design and other optimization techniques to reap the benefits of DiffSiam-Detach strategy without hurting convergence.

Chapter 5

Summary

5.1 Conclusion

We presented an un-/self-supervised disentangled representation learning algorithm, leveraging the inductive bias provided by Denoising Diffusion Probabilistic Model (DM)'s time-step. In particular, we observe an inherent connection between time-step and hidden modular attributes that generate data faithfully, *i.e.*, each incremental time-step in the forward diffusion process induces additional attribute loss, causing the failure of DM to reconstruct the original sample from the noisy ones. Hence we learn an expanding set of modular features as time-step increases to compensate for the cumulatively lost attributes by making up for DM's reconstruction error. Experimental results demonstrate superior performance on various classification benchmarks. Moreover, the learned feature enables counterfactual generation on CelebA, FFHQ and Bedroom datasets, validating its disentanglement quality.

5.2 Limitations and Future Research

We'll continue pushing forward unsupervised representation learning with diffusion models with the following potential research directions.

- Representation Improvement: While our model shows promising results, the learned representation can still be further improved. We empirically find that

certain attributes in a relative large scale, such as the shape of the objects and layouts of rooms, prove challenging to control solely through z. Instead, it is x_T that exhibits the ability to modify them. In order to achieve further disentanglement of the image representation, future research efforts could aim to develop a mechanism that enhances the dependency between these two variables, ultimately merging them.

- Leveraging Stable Diffusion for Rich Semantic Information: Recently, stable diffusion [57] is widely adopted in various generative tasks, revealing its superb ability to encode rich semantic information and produce high-quality images during decoding. Leveraging the substantial prior knowledge from stable diffusion offers the potential to create a broader representation learners capable of encoding real-world concepts with enhanced performance.

Bibliography

[1] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. xv, 3, 32, 37

[2] Zijian Zhang, Zhou Zhao, and Zhijie Lin. Unsupervised representation learning from pre-trained diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 35:22117–22130, 2022. xv, 2, 17, 24, 25, 32

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 1

[4] Tan Wang, Zhongqi Yue, Jianqiang Huang, Qianru Sun, and Hanwang Zhang. Self-supervised learning disentangled group representation as feature. *Advances in Neural Information Processing Systems*, 34:18225–18240, 2021. 1, 11

[5] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018. 1, 11

[6] Michel Besserve, Arash Mehrjou, Rémy Sun, and Bernhard Schölkopf. Counterfactuals uncover the modular structure of deep generative models. *ICLR*, 2020. 1

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 1, 8

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 6, 7, 8

[9] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2, 15

[10] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ICLR*, 2021. 2, 22

[11] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34: 8780–8794, 2021. 2, 20

[12] 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 preprint arXiv:2112.10741*, 2021. 2, 15

[13] Zhongqi Yue, Tan Wang, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Counterfactual zero-shot and open-set visual recognition. In *CVPR*, 2021. 2

[14] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, pages 881–889. PMLR, 2015. 2

[15] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[16] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017. 2

[17] 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*, pages 10619–10629, 2022. 2, 17

[18] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. 4, 8, 10, 20, 22, 30

[19] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 5, 9

[20] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 5

[21] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of the IEEE international conference on computer vision*, pages 37–45, 2015. 5

[22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[23] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 6, 7, 8

[24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 6, 7

[25] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010. 6

[26] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 7

[27] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020. 8

[28] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2840–2848, 2017. 8

[29] Ben Harwood, Vijay Kumar BG, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2821–2829, 2017. 8

[30] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 8

[31] 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*, pages 12310–12320. PMLR, 2021. 8

[32] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 9

[33] Zengyi Li, Yubei Chen, Yann LeCun, and Friedrich T Sommer. Neural manifold clustering and embedding. *arXiv preprint arXiv:2201.10000*, 2022. 8

[34] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1058–1067, 2017. 9

[35] 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*, pages 9650–9660, 2021. 9

[36] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 9

[37] 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*, pages 16000–16009, 2022. 9

[38] Florian Bordes, Randall Balestriero, and Pascal Vincent. High fidelity visualization of what your self-supervised representation knows about. *arXiv preprint arXiv:2112.09164*, 2021. 9

[39] Karl Ridgeway and Michael C Mozer. Learning deep disentangled embeddings with the f-statistic loss. *Advances in neural information processing systems*, 31, 2018. 11

[40] Xinqi Zhu, Chang Xu, and Dacheng Tao. Commutative lie group vae for disentanglement learning. In *ICML*, 2021. 11

[41] Kien Do and Truyen Tran. Theory and evaluation metrics for learning disentangled representations. In *International conference on learning representations*, 2020. 12

[42] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, 2019. 12, 14

[43] Diederik Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 13

[44] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 13

[45] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *International conference on learning representations*, 2017. 13

[46] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017. 13

[47] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017.

[48] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, 2018.

[49] Emilien Dupont. Learning disentangled joint continuous and discrete representations. *Advances in neural information processing systems*, 31, 2018. 13

[50] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 13

[51] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, 2016. 14

[52] Utkarsh Ojha, Krishna Kumar Singh, Cho-Jui Hsieh, and Yong Jae Lee. Elastic-infogan: Unsupervised disentangled representation learning in class-imbalanced data. In *Advances in neural information processing systems*, 2020. 14

[53] Zinan Lin, Kiran Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *International Conference on Machine Learning*, 2020.

[54] Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun Zeng. Do generative models know disentanglement? contrastive learning is all you need. *arXiv preprint arXiv:2102.10543*, 2021. 14

[55] 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*, pages 4401–4410, 2019. 14, 32

[56] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 14, 32, 37

[57] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 15, 16, 44

[58] 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. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[59] Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. *arXiv preprint arXiv:2303.06555*, 2023. 15

[60] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 15

[61] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. *arXiv preprint arXiv:2302.03011*, 2023.

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

[63] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20908–20918, 2023. 15

[64] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023.

[65] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 15

[66] Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pages 8867–8887. PMLR, 2022. 15

[67] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 15, 20, 21, 29

[68] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 15, 21

[69] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2435, 2022. 16

[70] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. *ICLR*, 2022. 16, 17

[71] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 16

[72] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 16

[73] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 16

[74] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 16

[75] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 16

[76] Dani Valevski, Matan Kalman, Yossi Matias, and Yaniv Leviathan. Unitune: Text-driven image editing by fine tuning an image generation model on a single image. *arXiv preprint arXiv:2210.09477*, 2022.

[77] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 16

[78] Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. More control for free! image synthesis with semantic diffusion guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 289–299, 2023. 16

[79] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022. 16

[80] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *arXiv preprint arXiv:2108.00946*, 2021. 16

[81] Dongxu Yue, Qin Guo, Munan Ning, Jiaxi Cui, Yuesheng Zhu, and Li Yuan. Chatface: Chat-guided real face editing via diffusion latent space manipulation. *arXiv preprint arXiv:2305.14742*, 2023. 17

[82] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *arXiv preprint arXiv:2203.09481*, 2022. 17

[83] Yingheng Wang, Yair Schiff, Aaron Gokaslan, Weishen Pan, Fei Wang, Christopher De Sa, and Volodymyr Kuleshov. Infodiffusion: Representation learning using information maximizing diffusion models. *arXiv preprint arXiv:2306.08757*, 2023. 17

[84] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 20, 21, 32

[85] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6:737–744, 1993. 22

[86] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010. 23

[87] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 23

[88] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 29

[89] Linus Ericsson, Henry Gouk, and Timothy M. Hospedales. How Well Do Self-Supervised Models Transfer? In *CVPR*, 2021. 29

[90] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 29

[91] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 29

[92] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 29

[93] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 29

[94] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, 2010. 29

[95] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 29

[96] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, Apr 2022. 29

[97] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008. 29

[98] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 29

[99] Sjoerd Van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are disentangled representations helpful for abstract visual reasoning? In *Advances in neural information processing systems*, 2019. 31

[100] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 32

[101] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 37

[102] Xiao Zhou, Weizhong Zhang, Hang Xu, and Tong Zhang. Effective sparsification of neural networks with global sparsity constraint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3599–3608, 2021. 37