THE UNIVERSITY OF CHICAGO

INTERPRETABLE UNSUPERVISED GENERATIVE LEARNING VIA FACTORIZED ARCHITECTURES AND STRUCTURED BOTTLENECKS

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

DEPARTMENT OF COMPUTER SCIENCE

BY
XIN YUAN

CHICAGO, ILLINOIS
AUGUST 2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Completing this PhD has been a profoundly challenging and enriching journey, one that has tested my limits and expanded my horizons in countless ways. It has been a period of intense learning, personal growth, and professional development, which would not have been possible without the support and encouragement of a group of remarkable individuals to whom I owe a great deal of gratitude.

First and foremost, I must extend my deepest appreciation to my advisor, Dr. Michael Maire. His invaluable mentorship, persistent guidance, and unwavering support have been pivotal throughout this journey. Dr. Maire's deep insights and dedicated involvement have been instrumental in shaping both the direction of my research and my growth as a researcher. His encouragement in times of challenge was as important as his academic guidance, and for that, I am immensely thankful.

I am also profoundly grateful to the members of my thesis committee: Dr. Rana Hanocka, Dr. Greg Shakhnarovich, and Dr. Anand Bhattad. Each has contributed to my project in unique ways, providing rigorous feedback and posing challenging questions that encouraged me to think more deeply and refine my work further. Their expertise and thoughtful engagement have greatly enhanced the quality of my research and my development as a researcher.

The financial support provided by the National Science Foundation under grant CNS-1956180 and the University of Chicago CERES Center has been crucial. These grants not only funded my research but also gave me the freedom to pursue ambitious projects and participate in important academic conferences, which enriched my PhD experience significantly.

I would like to extend a heartfelt thank you to my colleagues and friends from the lab, especially Pedro Savarese and Xiao Zhang. Our countless discussions have sparked new ideas, solved tough problems, and made the daily grind of research both enjoyable and rewarding. Their camaraderie and intellectual generosity have greatly enriched my time at the university.

A special acknowledgment goes to my family, whose love and unwavering support have been my anchor throughout this journey. Their endless encouragement and belief in my capabilities have been a constant source of strength and motivation. I am particularly thankful for their understanding during the many times my research demanded my focus and time.

Lastly, to my beloved Huiyu, thank you for your patience, understanding, and the countless sacrifices you have made. Your support has been a cornerstone of my PhD journey, providing me with stability and comfort during the most stressful periods. Your belief in my work and your unwavering love have been sources of immense strength.

This PhD journey has been more than just an academic pursuit; it has been a life-changing experience that has prepared me for the future in ways I could never have anticipated. For all this and more, I am deeply grateful to everyone who has been part of my PhD journey.

ABSTRACT

In this thesis, we propose an innovative paradigm for constructing generative models, fundamentally rethinking the conventional framework used in image generation and representation learning. Our approach centers around designing a domain-specific architecture that enables unified, unsupervised image generation and representation learning. This architecture incorporates a meticulously engineered bottleneck data structure, which is crafted with an acute understanding of the specific requirements of the task at hand, the characteristics of the data involved, and the computational constraints inherent to the problem. This bottleneck structure is pivotal, as it directly addresses the tasks to be solved by facilitating a learning process that generates useful outputs without reliance on direct supervision. This stands in stark contrast to traditional methodologies, which typically involve training large-scale foundation models in a self-supervised manner and subsequently fine-tuning them on annotated data for specific downstream tasks. Our proposed method eliminates the need for such fine-tuning and does not require annotated data at any stage of the pre-training process.

To demonstrate the effectiveness and robustness of our proposed design, we have conducted extensive validation across a variety of challenging tasks, each chosen to test different facets of the model under diverse experimental settings. These tests are crucial for proving the versatility and applicability of our approach in real-world scenarios, showcasing its potential to handle complex, unsupervised learning tasks in two experimental settings.

For the first experimental setting, we develop a neural network architecture which, trained in an unsupervised manner as a denoising diffusion model, simultaneously learns to both generate and segment images. Learning is driven entirely by the denoising diffusion objective, without any annotation or prior knowledge about regions during training. A computational bottleneck, built into the neural architecture, encourages the denoising network to partition an input into regions, denoise them in parallel, and combine the results. Our trained model generates both synthetic images and, by simple examination of its internal predicted partitions, a semantic segmentation of those images. Without any finetuning, we directly apply our unsupervised model to the downstream

task of segmenting real images via noising and subsequently denoising them.

For the second experimental setting, we cast multiview reconstruction from unknown pose as a generative modeling problem. From a collection of unannotated 2D images of a scene, our approach simultaneously learns both a network to predict camera pose from 2D image input, as well as the parameters of a Neural Radiance Field (NeRF) for the 3D scene. To drive learning, we wrap both the pose prediction network and NeRF inside a Denoising Diffusion Probabilistic Model (DDPM) and train the system via the standard denoising objective. Our framework requires the system accomplish the task of denoising an input 2D image by predicting its pose and rendering the NeRF from that pose. Learning to denoise thus forces the system to concurrently learn the underlying 3D NeRF representation and a mapping from images to camera extrinsic parameters. To facilitate the latter, we design a custom network architecture to represent pose as a distribution, granting implicit capacity for discovering view correspondences when trained end-to-end for denoising alone. This technique allows our system to successfully build NeRFs, without pose knowledge, for challenging scenes where competing methods fail. At the conclusion of training, our learned NeRF can be extracted and used as a 3D scene model; our full system can be used to sample novel camera poses and generate novel-view images.

Extensive experiments conducted as part of this thesis demonstrate the profound capability of our proposed factorized architecture and its integral structured computational bottleneck to address and resolve classical challenges in the field of computer vision, doing so end-to-end by purely learning to generate from unlabeled data. These experimental evaluations were rigorous and meticulously designed to test the versatility and robustness of our model under various scenarios, showcasing its ability to operate effectively across a broad spectrum of conditions without any dependency on labeled datasets. Specifically, the results of our experiments reveal that our model not only accomplishes accurate unsupervised image segmentation but also excels in generating high-quality synthetic images. This is evidenced across multiple datasets, where the model consistently performs with high precision and reliability, thereby indicating its suitability for diverse real-world applications. The ability of our model to segment images without supervision is particularly noteworthy, as it demonstrates a significant leap in the capability of generative models to understand and interpret complex visual data autonomously. Moreover, our research marks a pioneering advance in the field by being potentially the first to successfully tackle unsupervised pose estimation and 3D reconstruction within a diffusion-based framework for 360-degree scenes. This achievement is particularly significant, as it addresses a long-standing challenge in computer vision—achieving reliable 3D understanding from 2D inputs in an unsupervised manner. Our approach not only estimates the pose but also reconstructs the 3D geometry of the scene without any prior knowledge or external annotations, paving the way for new applications and improvements in areas such as virtual reality, augmented reality, and robotic navigation. These findings not only validate the efficacy of our novel generative architecture but also underscore its potential to transform the landscape of unsupervised learning in computer vision, opening up new avenues for research and application that were previously thought to be reliant on extensive labeled data. The success of these experiments thus provides a robust foundation for further exploration and development of unsupervised learning technologies in image and scene understanding.

CHAPTER 1

INTRODUCTION

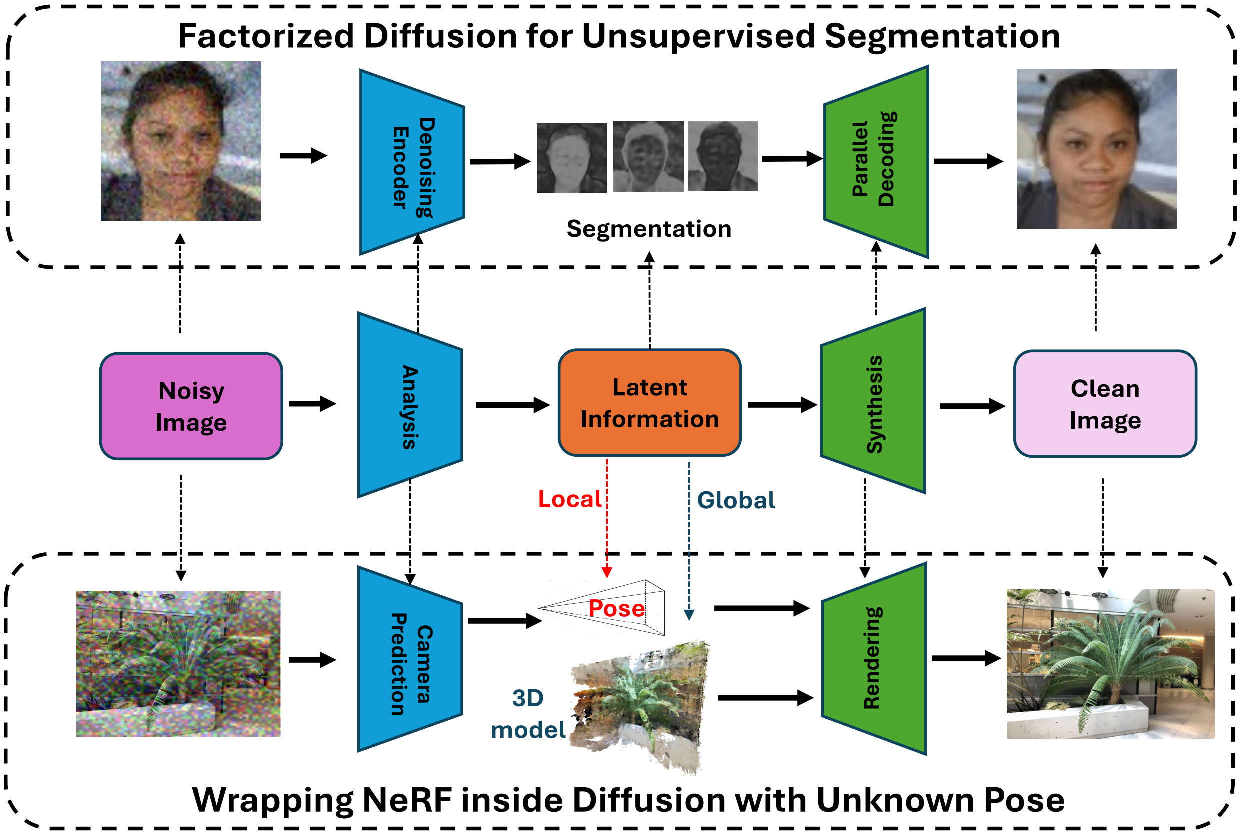


Figure 1.1: The proposed universal architectural design of this thesis. ***Middle:*** An encoder analyzes noisy images and generates latent information through a structured computational bottleneck, which can be interpreted by a synthesis module to produce clean reconstructions. By customizing the architectures, our model can learn unsupervised representations while generating high-quality outputs for different tasks: ***Top:*** A factorized diffusion architecture for unsupervised segmentation, which partitions the denoising network within a diffusion model into an unsupervised region mask generator and parallel per-region decoders. ***Bottom:*** Wrapping NeRF inside diffusion which consists of a camera prediction encoder to generate poses as local latent information. A globally shareable 3D model is optimized jointly through the decoder-like rendering system, driven by the denoising objective from multiple 2D images.

Supervised deep learning has been instrumental in driving significant advancements across a broad spectrum of computer vision tasks. The power of discriminative representations learned through supervised methods has fundamentally transformed the field, leading to remarkable achievements in various domains. Notably, supervised learning has catalyzed progress in image classification, as evidenced by pioneering works such as those by [Deng et al., 2009, Simonyan and Zisserman, 2015, He et al., 2016, Huang et al., 2017], which introduced and refined deep convolutional neural networks, setting new benchmarks in accuracy and efficiency. Similar advancements have been extended to object detection [Girshick et al., 2014, Redmon et al., 2016, Liu et al., 2016], where innovative approaches have led to more robust and faster detection algorithms. This progress is paralleled in the fields of semantic and instance segmentation, with seminal works [Long et al., 2015, He et al., 2017] to precisely delineate objects at both pixel and instance levels, thereby improving the systems' ability to interpret and interact with visual environments in a more meaningful way.

The development of these advancements has been the strategy of supervised pre-training over large-scale datasets, which yields rich, useful visual features that push state-of-the-art performance across these tasks. The success of this approach demonstrates the value of comprehensive and diverse datasets like ImageNet [Deng et al., 2009], which provide the foundational knowledge necessary for training highly effective models. However, despite these benefits, the reliance on extensively annotated datasets introduces significant challenges. The requirement for fine-grained labeling, necessary for training accurate models, escalates in cost and complexity as the size of the dataset increases. This scaling issue is exacerbated in tasks requiring detailed annotation, such as precise object segmentation or the identification of subtle nuances in large sets of images, making the process not only cost-prohibitive but also time-consuming and labor-intensive. This burgeoning need for vast amounts of labeled data poses a critical bottleneck in the scalability of supervised learning frameworks, prompting a growing interest in alternative methodologies that can bypass the intensive demands of manual annotation. As such, there is an increasing emphasis on exploring unsupervised and semi-supervised learning paradigms, which aim to reduce dependency on labeled data while still leveraging the underlying structure and information present within unlabeled datasets to achieve comparable or even superior performance. This shift represents a pivotal evolution in the approach to training deep learning models, aiming to maintain the momentum of innovation in computer vision while addressing the practical limitations imposed by dataset annotation requirements.

This growing need for high-quality labeled datasets alongside the complexity and costs associated with their creation motivates a significant shift towards the development of large-scale foundation models that do not require any annotated data for pre-training [Caron et al., 2019, Doersch et al., 2015, Zhang et al., 2016, Larsson et al., 2017, He et al., 2020, Chen et al., 2020a,b]. Such models offer a promising direction by potentially reducing the reliance on costly labeled data while still enabling powerful, scalable learning systems.

With the recent rapid advancements in deep generative models [Kingma and Welling, 2014, Goodfellow et al., 2014, Xu et al., 2018, Zhang et al., 2017, van den Oord et al., 2016, Li and Malik, 2018, Ho et al., 2020, Song et al., 2021, Rombach et al., 2022], a new frontier in representation learning has opened up, illustrating that image generation can not only facilitate realistic image synthesis but also serve as a viable proxy task for capturing high-level semantic information. This approach leverages the inherent capabilities of generative models to understand and recreate the distribution of input data, thus learning valuable features that can be applied across a variety of tasks. Several pioneering research efforts have begun to explore how representation learning can be effectively integrated within the framework of generative models. For instance, Zhang *et al.* [Zhang et al., 2023] have explored the potential of incorporating spectral clustering techniques to dissect and harness the rich information contained within a pre-trained stable diffusion model. Their work demonstrates how advanced clustering methods can be used to identify and extract meaningful patterns and features from the model, which are crucial for enhancing its generative and discriminative capabilities. Similarly, Du *et al.* [Du et al., 2023] have adopted a Low-Rank Adaptation (LoRA) strategy on a labeled subset to refine the performance of various generative models, including diffusion models and GANs. By focusing on extracting scene intrinsic maps from these models, their approach highlights the adaptability and utility of generative frameworks in understanding and replicating complex scene dynamics. This adaptation enhances the model's understanding of fundamental scene characteristics, such as lighting, geometry, and texture. These examples underscore the potential of generative models not just as tools for image creation but as robust platforms for deep, nuanced learning of visual representations. By moving towards systems that can autonomously learn from vast amounts of unlabeled data, researchers are paving the way for more efficient, scalable, and cost-effective solutions in the field of computer vision. This shift is not merely a response to the limitations of labeled datasets but also an embrace of the richer, more diverse learning opportunities that generative models provide. As these technologies continue to evolve, they promise to unlock new capabilities and applications that will further enhance our ability to analyze, interpret, and interact with visual information in innovative ways.

Unfortunately, while self-supervised learning approaches [Caron et al., 2019, Doersch et al., 2015, Zhang et al., 2016, Larsson et al., 2017, He et al., 2020, Chen et al., 2020a,b] have proven effective as feature extractors, leveraging extremely large training sets or extended training periods, they cannot inherently directly address and solve specific end tasks on their own. This limitation underscores a critical challenge within the domain of self-supervised learning, where despite significant advances in feature extraction and representation learning, the transition to practical applications remains dependent on further supervised intervention. Many self-supervised methods [He et al., 2020, Chen et al., 2020a,b], therefore, necessitate a subsequent phase of supervised fine-tuning, such as linear probing, to adapt these pre-trained networks to specific downstream tasks. This step is essential to refine the broad, generalizable features extracted during the self-supervised phase into task-specific models that perform effectively on targeted applications. This requirement for fine-tuning highlights a fundamental gap in self-supervised approaches, where the initial learning phase, despite its sophistication and breadth, does not culminate in a standalone solution ready for direct application. In the realm of generative representation learning, the situation presents similar challenges. Existing works [Baranchuk et al., 2022, Chen et al., 2023b, Zhang et al., 2023, Du et al., 2023] predominantly rely on large-scale pre-trained generative models which are capable of synthesizing and manipulating complex data distributions, potentially leading to the extraction of interpretable features. However, while these features provide a rich representation of the data, fine-tuning is necessary to address end tasks. Consequently, even with advanced generative models, additional steps involving post-processing or fine-tuning with further labeled data are indispensable. This requirement not only extends the development cycle but also adds complexity and resource demands, particularly in scenarios where labeled data is scarce, expensive, or difficult to procure. This continued reliance on labeled data for fine-tuning postulates a significant limitation in the autonomous capabilities of both self-supervised and generative learning paradigms. Thus, while these innovative learning strategies offer substantial improvements over traditional generative methods in terms of extracting versatile features, the complete pipeline from training to practical application often still hinges on incorporating supervised learning elements to achieve the desired performance on specific tasks. This dependency on post-processing or fine-tuning raises an ongoing challenge in the field: the development of methodologies that can truly operate independently of labeled data from scratch, thereby realizing the full potential of unsupervised and generative learning techniques.

In parallel with the progress of various training paradigms for generation and representation learning, deep network architecture design is a foundational element in increasing the model capability. It dictates how a network processes information, learns from data, and represents knowledge, fundamentally influencing its ability to understand and tackle complex tasks. A well-crafted architecture can enhance the learning capacity, generalize well across different datasets, and improve overall performance, making it a cornerstone in the development of intelligent systems. For example, in Denoising Diffusion Probabilistic Model (DDPM) [Ho et al., 2020], the denoising UNet stands out as a core architectural design. This specialized version of the UNet architecture [Ronneberger et al., 2015] is tailored for image denoising, learning to map noisy images to clean ones by minimizing the difference between predicted and ground truth images. The denoising UNet's encoding-decoding pathways, along with skip connections, play a crucial role in preserving spatial information, effectively removing noise while retaining essential image details. This architecture is key to recovering the image distribution during DDPM training, although additional steps such as fine-tuning with annotations may be required for comprehensive representation learning. For another recently proposed application of Neural Radiance Fields (NeRF) [Mildenhall et al., 2020], the architectural design is centered around representing the 3D scene and rendering new views. NeRF typically employs a multi-layer perceptron (MLP) as its core architecture, comprising fully connected layers with non-linear activations. This setup allows the network to learn intricate mappings from input coordinates to output color and density values. To synthesize novel views, NeRF utilizes ray marching along camera rays, evaluating the network along each ray to estimate color and opacity. However, NeRF requires pose information, and existing architectures do not support training from unknown poses for complex scenes with only training once.

In this thesis, we propose an innovative approach within the context of generative models by designing a domain-specific architecture that incorporates specialized, structured components tailored to specific tasks or domains. This architecture is intended to enhance traditional generative model designs with the ability to directly learn useful representations in an unsupervised manner, thereby bypassing the limitations often associated with standard models that require extensive labeled datasets. To achieve this, we adopt an encoder-decoder architecture enhanced with structured representations and denoising diffusion objectives, allowing the system to learn powerful representations for a variety of tasks. The encoder component of this architecture is designed to take a noisy image as input and process it through multiple layers, extracting and condensing the information into a latent representation. This latent space is crafted through a carefully designed structured computational bottleneck, which plays a crucial role in ensuring that the representations are highly informative and relevant for the tasks at hand. The structured bottleneck is a pivotal element of our architecture, designed to impose specific constraints on the information flow within the network. This forces the encoder to focus on extracting only the most essential features from the input data, which are necessary for successful task performance. By doing so, it enhances the efficiency and effectiveness of the representation learning process. Following the encoder, the decoder component takes over by taking the structured latent representation and working to reconstruct the original input image. This reconstruction is not a mere replication of the input but a reassembly that incorporates and reflects the structured information encapsulated in the latent representation. The decoder's ability to accurately reconstruct the original image from the modified latent space serves as a testament to the quality and utility of the learned representations. The entire encoder-decoder system is trained end-to-end with a focus on minimizing a denoising loss function. This training approach ensures that both the encoder and decoder are optimized in unison, allowing the structured computational bottleneck to effectively shape the learning process and directly generate useful, task-specific representations. The training strategy is specifically designed to allow for continuous adaptation and improvement of the bottleneck's structure, making the system increasingly effective over time. Furthermore, the general design of our system is visualized in Figure 1.1 (middle), which provides a schematic representation of the encoder-decoder flow and highlights the key components and their interactions. By customizing this architecture for different domains or tasks, our model is not only capable of learning unsupervised representations but also excels at generating high-quality outputs tailored to the specific requirements of each task. This adaptability and versatility demonstrate the potential of our proposed architecture to transform the landscape of generative modeling, making it a powerful tool for a wide range of applications where unsupervised learning is desirable or necessary.

- **Factorized Diffusion for Segmentation (Figure 1.1, top):** In the proposed architecture, a specialized denoising encoder is utilized to process the input image, transforming it into a latent segmentation representation. This representation captures essential features and structural information of the input, optimized for segmenting the image into distinct, meaningful regions. This latent segmentation representation is not just a compressed form of the input image; it is a refined, structured output that encapsulates the core aspects of the image necessary for precise segmentation. The encoder leverages advanced denoising techniques to ensure that this representation is both clean and informative, making it ideally suited for the segmentation task. This involves sophisticated neural network layers that apply a series of transformations, each designed to refine the information and enhance its relevance to the task.

Once this latent representation is formed, it is fed into a parallel decoding scheme. This innovative aspect of the architecture allows for simultaneous, multi-path decoding, where each path is responsible for reconstructing a specific aspect of the original image. This parallel decoding approach enhances the accuracy and quality of the output by enabling specialized handling of different features or regions of the image. The parallel decoders work in synergy, each contributing to the final reconstructed output by focusing on different layers or aspects of the latent representation. Some paths might focus on delineating edges and boundaries, while others might enhance texture or color consistency within segmented regions. This method allows for a comprehensive and nuanced reconstruction of the image, with each segment clearly defined and distinct from its neighbors.

Furthermore, the parallel decoding scheme incorporates feedback mechanisms between the paths, enabling them to adjust and refine their outputs based on the results of other paths. This inter-path communication ensures that the final segmentation is not only accurate but also cohesive, with all segments well-integrated and visually consistent, making it a powerful tool for a variety of applications that require precise image analysis and generation.

- **Wrapping NeRF inside Diffusion with Unknown Pose (Figure 1.1, bottom):** In the proposed architecture, a sophisticated camera prediction system plays a central role by generating camera pose as local latent information in an entirely unsupervised manner. This system is designed to autonomously determine the orientation and position of the camera from which an image was captured, using only the image data itself without the need for pre-labeled pose information. This capability is particularly crucial for tasks involving 3D reconstruction from 2D images where accurate pose estimation is essential for synthesizing consistent and realistic three-dimensional views. The camera prediction system employs advanced machine learning techniques that enable it to infer these pose parameters by analyzing patterns and geometrical consistency across a sequence of images. It leverages deep neural networks trained to recognize and interpret the subtle cues that indicate camera position and orientation, such as vanishing points, object sizes, and relative positions in different images. This training is enhanced by a multi-view learning approach, where the system examines multiple images of the same scene taken from different angles, learning to understand how changes in camera pose affect the appearance of objects and scenes.

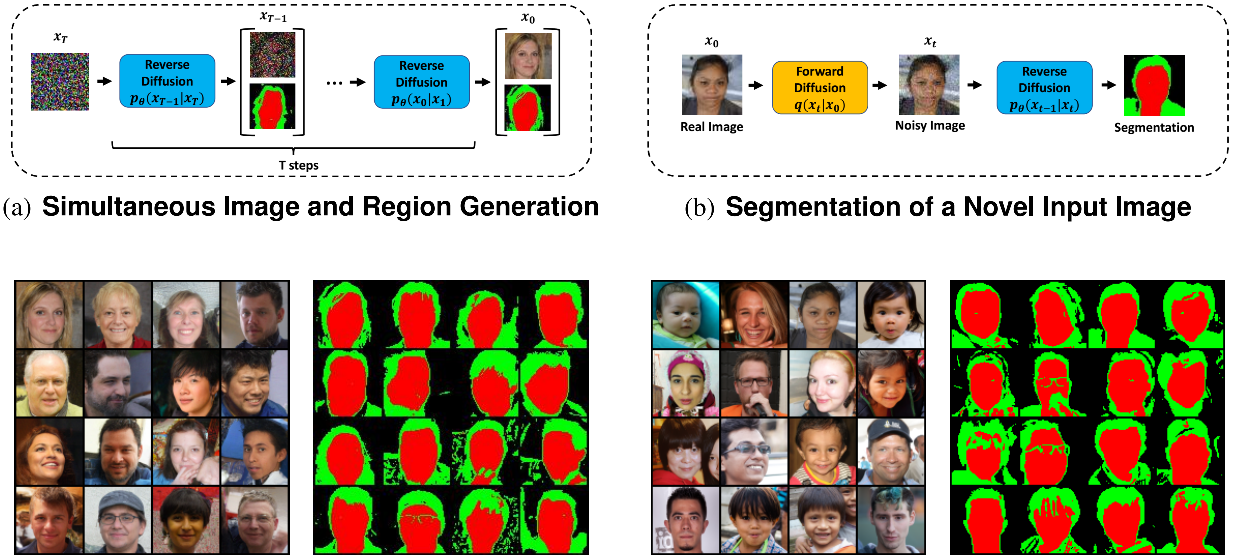
Once the camera poses are predicted, the system uses this information to sample differentiable rays that are crucial for generating global 3D latent information. These rays, emanating from the camera's viewpoint through the scene, interact with the 3D environment, allowing the system to compute intersections with virtual objects and thereby reconstruct the three-dimensional structure of the scene. This process involves a sophisticated rendering technique that simulates how light travels through space and interacts with surfaces, capturing the essence of the scene's geometry and appearance.

This joint multi-view learning and rendering framework is integral to our system. It not only enhances the accuracy of pose prediction by providing a rich context for understanding camera dynamics but also facilitates the extraction and integration of comprehensive 3D information from multiple viewpoints without pose annotations. By combining these views, the system can construct a detailed and accurate 3D model of the scene, enriched by the nuances that each separate view provides. The entire process is trained in an end-to-end manner, ensuring that both local latent information (camera pose) and global latent information (3D structure) are continuously refined and optimized through iterative learning. This results in a robust system capable of performing complex tasks such as photorealistic 3D reconstruction from a series of 2D images, all conducted in an unsupervised framework. The potential applications of this technology are vast, ranging from virtual reality and film production to architectural modeling and archaeological reconstruction, providing tools that can transform flat images into realistic, interactive 3D experiences.

CHAPTER 2

FACTORIZED DIFFUSION ARCHITECTURES FOR UNSUPERVISED IMAGE GENERATION AND SEGMENTATION

2.1 Introduction


x_T
Reverse Diffusion $p_\theta(x_{T-1}|x_T)$
x_{T-1}
...
Reverse Diffusion $p_\theta(x_0|x_1)$
x_0
T steps
x_0
Real Image
Forward Diffusion $q(x_t|x_0)$
x_t
Noisy Image
Reverse Diffusion $p_\theta(x_{t-1}|x_t)$
Segmentation

(a) **Simultaneous Image and Region Generation** (b) **Segmentation of a Novel Input Image**

(c) **Generated Images** (d) **Generated Regions** (e) **Real Images** (f) **Segmentations**

Figure 2.1: **Unifying image generation and segmentation.** (a) We design a denoising diffusion model with a specific architecture that couples region prediction with spatially-masked diffusion over predicted regions, thereby generating both simultaneously. (b) An additional byproduct of running our trained denoising model on an arbitrary input image is a segmentation of that image. Using a model trained on FFHQ [Karras et al., 2019], we achieve both high quality synthesis of images and corresponding semantic segmentations (c-d), as well as the ability to accurately segment images of real faces (e-f). Segmenting a real image is fast, requiring only one forward pass (one denoising step).

Supervised deep learning yields powerful discriminative representations, and has fundamentally advanced many computer vision tasks, including image classification [Deng et al., 2009, Simonyan and Zisserman, 2015, He et al., 2016, Huang et al., 2017], object detection [Girshick et al., 2014, Redmon et al., 2016, Liu et al., 2016], and semantic and instance segmentation [Long et al., 2015, He et al., 2017, Kirillov et al., 2023]. Yet, annotation efforts [Deng et al., 2009], especially those involving fine-grained labeling for tasks such as segmentation [Lin et al., 2014], can become prohibitively expensive to scale with increasing dataset size. This motivates an ongoing revolution in self-supervised methods for visual representation learning, which do not require any annotated data during a large-scale pre-training phase [Caron et al., 2019, Doersch et al., 2015, Zhang et al., 2016, Larsson et al., 2017, He et al., 2020, Chen et al., 2020a,b]. However, many of these approaches, including those in the particularly successful contrastive learning paradigm [He et al., 2020, Chen et al., 2020a,b], still require supervised fine-tuning (*e.g.,* linear probing) on labeled data to adapt networks to downstream tasks such as classification [He et al., 2020, Chen et al., 2020a] or segmentation [Caron et al., 2021, Zhang and Maire, 2020].

In parallel with the development of self-supervised deep learning, rapid progress on a variety of frameworks for deep generative models [Kingma and Welling, 2014, Goodfellow et al., 2014, Xu et al., 2018, Zhang et al., 2017, van den Oord et al., 2016, Li and Malik, 2018, Ho et al., 2020, Song et al., 2021, Rombach et al., 2022] has lead to new systems for high-quality image synthesis. This progress inspires efforts to explore representation learning within generative models, with recent results suggesting that image generation can serve as a good proxy task for capturing high-level semantic information, while also enabling realistic image synthesis.

Building upon generative adversarial networks (GANs) [Goodfellow et al., 2014] or variational autoencoders (VAEs) [Kingma and Welling, 2014], InfoGAN [Chen et al., 2016] and Deep InfoMax [Hjelm et al., 2019] demonstrate that generative models can perform image classification without any supervision. PerturbGAN [Bielski and Favaro, 2019] focuses on a more complex task, unsupervised image segmentation, by forcing an encoder to map an image to the input of a pre-trained generator so that it synthesizes a composite image that matches the original input image. However, here training is conducted in two stages and mask generation relies on knowledge of predefined object classes.

Denoising diffusion probabilistic models (DDPMs) [Ho et al., 2020] also achieve impressive performance in generating realistic images. DatasetDDPM [Baranchuk et al., 2022] investigates the intermediate activations from the pre-trained U-Net [Ronneberger et al., 2015] network that approximates the Markov step of the reverse diffusion process in DDPM, and proposes a simple semantic segmentation pipeline fine-tuned on a few labeled images. In spite of this usage of labels, DatasetDDPM demonstrates that high-level semantic information, which is valuable for downstream vision tasks, can be extracted from pre-trained DDPM U-Net. Diff-AE [Preechakul et al., 2022] and PADE [Zhang et al., 2022] are recently proposed methods for representation learning by reconstructing images in the DDPM framework. However, their learned representations are in the form of a latent vector containing information applicable for image classification.

In contrast to all of these methods, we demonstrate a fundamentally new paradigm for unsupervised visual representation learning with generative models: constrain the architecture of the model with a structured bottleneck that provides an interpretable view of the generation process, and from which one can simply read off desired latent information. This structured bottleneck does not exist in isolation, but rather is co-designed alongside the network architecture preceding and following it. The computational layout of these pieces must work together in a manner that forces the network, when trained from scratch for generation alone, to populate the bottleneck data structure with an interpretable visual representation.

We demonstrate this concept in the scenario of a DDPM for image generation and the selection of semantic segmentation as the interpretable representation to be read from the bottleneck. Thus, we frame unsupervised image segmentation and generation in a unified system. Moreover, experiments demonstrate that domain-specific bottleneck design not only allows us to accomplish an end task (segmentation) for free, but also boosts the quality of generated samples. This challenges the assumption that generic architectures (*e.g.,* Transformers) alone suffice; we find synergy by organizing such generic building blocks into a factorized architecture which generates different image regions in parallel.

Figure 3.2 provides an overview of our setting alongside example results, while Figure 3.1 illustrates the details of our DDPM architecture which are fully presented in Section 3.3. This architecture constrains the computational resources available for denoising in a manner that encourages learning of a factorized model of the data. Specifically, each step of the DDPM has the ability to utilize additional inference passes through multiple copies of a subnetwork if it is willing to decompose the denoising task into parallel subproblems. The specific decomposition strategy itself must be learned, but, by design, is structured in a manner that reveals the solution to our target task of image segmentation. We summarize our contributions as three-fold:

- **Unified learning of generation and segmentation.** We train our new DDPM architecture once, obtaining a model directly applicable to two different tasks with zero modification or fine-tuning: image generation and image segmentation. Segmenting a novel input image is fast, comparable in speed to any system using a single forward pass of a U-Net [Ronneberger et al., 2015] like architecture.

- **Unsupervised segmentation for free.** Our method automatically learns meaningful regions (*e.g.,* foreground and background), guided only by the DDPM denoising objective; no extra regularization terms, no use of labels.

- **Higher quality image synthesis.** Our model generates higher-quality images than the baseline DDPM, as well as their corresponding segmentations simultaneously. We achieve excellent quantitative and qualitative results under common evaluation protocols (Section 2.4).

Beyond improvements to image generation and segmentation, our work is a case study of a new paradigm for using generation as a learning objective, in combination with model architecture as a constraint. Rather than viewing a pre-trained generative model as a source from which to extract and repurpose features for downstream tasks, design the model architecture in the first place so that, as a byproduct of training from scratch to generate, it also learns to perform the desired task.

2.2 Related Work

Image Segmentation. Generic segmentation, which seeks to partition an image into meaningful regions without prior knowledge about object categories present in the scene, is a longstanding challenge for computer vision. Early methods rely on combinations of hand-crafted features based on intensity, color, and texture cues [Canny, 1986, Martin et al., 2004], clustering algorithms [Shi and Malik, 2000], and a duality between closed contours and the regions they bound [Arbeláez et al., 2011]. Deep learning modernized the feature representations used in these pipelines, yielding systems which, trained with supervision from annotated regions [Martin et al., 2001], reach near human-level accuracy on predicting and localizing region boundaries [Bertasius et al., 2015, Shen et al., 2015, Xie and Tu, 2015, Kokkinos, 2016].

Semantic segmentation, which assigns a category label to each pixel location in image, has been similarly revolutionized by deep learning. Here, the development of specific architectures [Long et al., 2015, Ronneberger et al., 2015, Hariharan et al., 2015] enabled porting of approaches for image classification to the task of semantic segmentation.

Recent research has refocused on the challenge of learning to segment without reliance on detailed annotation for training. Hwang et al. [2019] combine two sequential clustering modules for both pixel-level and segment-level to perform this task. Ji et al. [2019] and Ouali et al. [2020] follow the concept of mutual information maximization to partition pixels into two segments. Savarese et al. [2021] further propose a learning-free adversarial method from the information theoretic perspective, with the goal of minimizing predictability among different pixel subsets. Note that even completely unsupervised foreground/background segmentation is a non-trivial task. Liu et al. [2021], a recent advance in this regime, produces similar region mask output, yet depends entirely upon motion cues from video for training. We achieve such unsupervised learning from static images alone.

Learning Segmentation in Generative Models. Previous generative model-based approaches learn semantic segmentation by perturbing [Bielski and Favaro, 2019] or redrawing [Chen et al., 2019] generated foreground and background masks. Despite good performance, these methods apply only to two-class partitions and require extra loss terms based upon object priors in training datasets.

Denoising diffusion probabilistic models (DDPMs) [Ho et al., 2020] achieve state-of-the-art performance in generating realistic images. Their noise schedule in training may offer advantages for scaling up models in a stable manner. Recent works [Baranchuk et al., 2022, Preechakul et al., 2022, Zhang et al., 2022] explore representation learning capability in DDPMs. DatasetDDPM [Baranchuk et al., 2022] examines few-shot segmentation with pre-trained diffusion models, but requires human labels to train a linear classifier. With the default U-Net architecture [Ronneberger et al., 2015], it loses the efficiency and flexibility of generating image and masks in a single-stage manner. Diff-AE [Preechakul et al., 2022] and PADE [Zhang et al., 2022] perform representation learning driven by a reconstruction objective in the DDPM framework. Unfortunately, their learned latent vectors are not applicable to more challenging segmentation tasks and they require a pre-trained interpreter to perform downstream image classification.

DiffuMask [Wu et al., 2023] takes a pre-trained Stable Diffusion model [Rombach et al., 2022], which is built using large-scale text-to-image datasets (and thus solves a far less challenging problem), and conducts a post-hoc investigation on how to extract segmentation from its attention maps. Neither our system, nor the baseline DDPM to which we compare, makes use of such additional information. Furthermore, DiffuMask does not directly output segmentation; it is basically a dataset generator, which produces generated images and pseudo labels, which are subsequently used to train a separate segmentation model. Our method, in contrast, is both completely unsupervised and provides an end-to-end solution by specifying an architectural design in which training to generate reveals segmentations as a bonus.

MAGE [Li et al., 2022] shares with us a similar motivation of framing generation and representation learning in a unified framework. However, our approach is distinct in terms of both (1) task: we tackle a more complex unsupervised segmentation task (without fine-tuning) instead of image classification (with downstream fine-tuning), and (2) design: ‘masks’ play a fundamentally different role in our system. MAGE adopts an MAE [He et al., 2022]-like masking scheme on input data, in order to provide a proxy reconstruction objective for self-supervised representation learning. Our use of region masks serves a different purpose, as they are integral components of the model being learned and facilitate factorization of the image generation process into parallel synthesis of different segments.

BlobGAN [Epstein et al., 2022] is a generative model for creating images with fine-grained control over the spatial arrangement of content. It leverages blob-like components instead of accurate region masks as basic building blocks for the synthesis process, allowing for intuitive content manipulation. In the generative modeling space, BlobGAN serves a different purpose than our method: BlobGAN excels in scenarios requiring explicit spatial control and interactive editing, while our factorized diffusion approach provides a framework for learning high-quality image generation and segmentation.

2.3 Factorized Diffusion Models

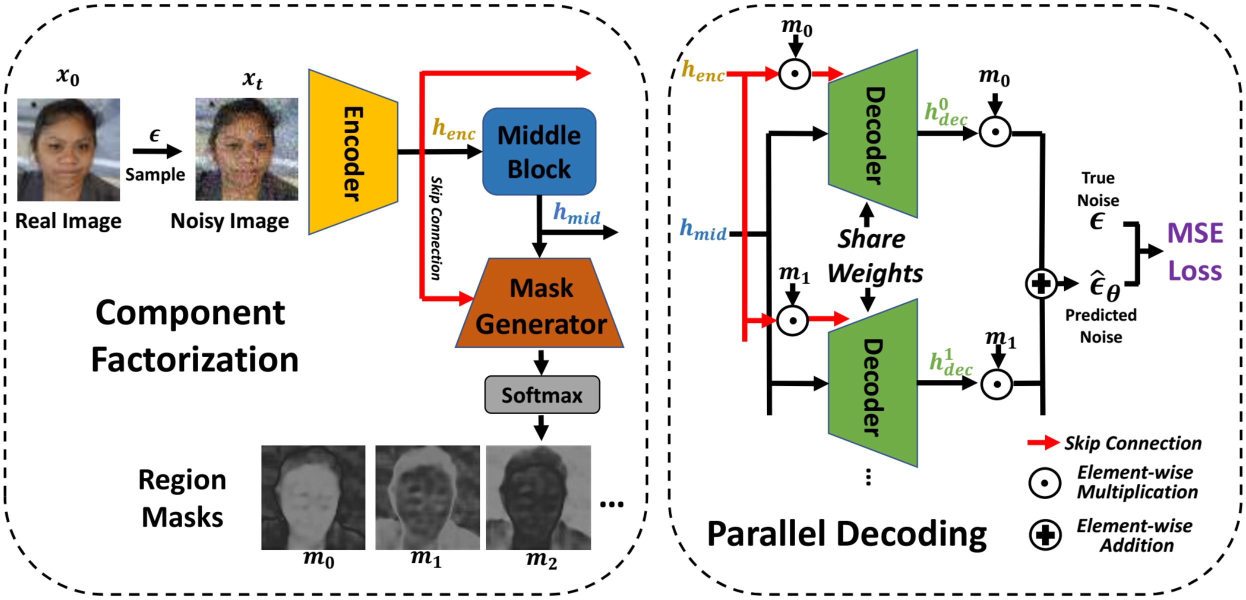


Figure 2.2: **Factorized diffusion architecture.** Our framework restructures the architecture of the neural network within a DDPM [Ho et al., 2020] so as to decompose the image denoising task into parallel subtasks. All modules are end-to-end trainable and optimized according to the same denoising objective as DDPM. ***Left: Component factorization.*** An *Encoder*, equivalent to the first half of a standard DDPM U-Net architecture, extracts features $\boldsymbol{h}_{enc}$. A common *Middle Block* processes *Encoder* output into shared latent features $\boldsymbol{h}_{mid}$. Note that *Middle Block* and $\boldsymbol{h}_{mid}$ exist in the standard denoising DDPM U-Net by default. We draw it as a standalone module for a better illustration of the detailed architectural design. A *Mask Generator*, structured as the second half of a standard U-Net receives $\boldsymbol{h}_{mid}$ as input, alongside all encoder features $\boldsymbol{h}_{enc}$ injected via skip connections to layers of corresponding resolution. This later network produces a soft classification of every pixel into one of K region masks, $\boldsymbol{m_0}, \boldsymbol{m_1}, ..., \boldsymbol{m_K}$. ***Right: Parallel decoding.*** A *Decoder*, also structured as the second half of a standard U-Net, runs separately for each region. Each instance of the *Decoder* receives shared features $\boldsymbol{h}_{mid}$ and a masked view of encoder features $\boldsymbol{h}_{enc} \odot \boldsymbol{m_i}$ injected via skip connections to corresponding layers. Decoder outputs are masked prior to combination. Though not pictured, we inject timestep embedding $\boldsymbol{t}$ into the *Encoder*, *Mask Generator*, and *Decoder*.

Figure 3.1 illustrates the overall architecture of our system, which partitions the denoising network within a diffusion model into an unsupervised region mask generator and parallel per-region decoders.

2.3.1 Unsupervised Region Factorization

To simultaneously learn representations for both image generation and unsupervised segmentation, we first design the region mask generator based on the first half (encoder) of a standard DDPM U-Net. We obtain input $\boldsymbol{x}_t$, a noised version of $\boldsymbol{x}_0$, via forward diffusion:

$$\begin{aligned} q(\boldsymbol{x}_t|\boldsymbol{x}_0) &:= \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)I), \\ \boldsymbol{x}_t &= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(0,1), \end{aligned} \tag{2.1}$$

where $\alpha_t = 1 - \beta_t, \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_t$.

In addition to the encoder half of the U-Net, we instantiate a middle block consisting of layers operating on lower spatial resolution features. Parameterizing these subnetworks as θ_{enc} and θ_{mid}, we extract latent representations:

$$\boldsymbol{h}_{enc} = \theta_{enc}(\boldsymbol{x}_t, t), \tag{2.2}$$

$$\boldsymbol{h}_{mid} = \theta_{mid}(\boldsymbol{h}_{enc}, t) \tag{2.3}$$

where $\boldsymbol{h}_{enc}$ encapsulates features at all internal layers of θ_{enc}, for subsequent use as inputs, via skip connections, to corresponding layers of decoder-style networks (second half of a standard U-Net).

We instantiate a mask generator, θ_{mask}, as one such decoder-style subnetwork. A softmax layer produces an output tensor with K channels, representing K different regions in image $\boldsymbol{x}_0$:

$$\boldsymbol{m}_k = \theta_{mask}(\boldsymbol{h}_{mid}, \boldsymbol{h}_{enc}, t) \tag{2.4}$$

Following a U-Net architecture, $\boldsymbol{h}_{enc}$ feeds into θ_{mask} through skip-connections.

2.3.2 Parallel Decoding Through Weight Sharing

We aim to extend a standard DDPM U-Net decoder θ_{dec} to consider region structure during generation. One simple design is to condition on $\boldsymbol{m} = \{\boldsymbol{m_0}, \boldsymbol{m_1}, ...\}$ by concatenating it with input $\boldsymbol{h}_{mid}$ and $\boldsymbol{h}_{enc}$ along the channel dimension:

$$\hat{\boldsymbol{\epsilon}} = \theta_{dec}(\text{concat}[\boldsymbol{h}_{mid}, \boldsymbol{m}], \text{concat}[\boldsymbol{h}_{enc}, \boldsymbol{m}], t), \tag{2.5}$$

where $\boldsymbol{h}_{mid}$ and $\boldsymbol{h}_{enc}$ are generated from Eq. 2.2 and Eq. 2.3. We downsample $\boldsymbol{m}$ accordingly to the same resolution as $\boldsymbol{h}_{mid}$ and $\boldsymbol{h}_{enc}$ at different stages. However, such a design significantly modifies (*e.g.,* channel sizes) the original U-Net decoder architecture. Moreover, conditioning with the whole mask representation may also result in a trivial solution that simply ignores region masks.

To address these issues, we separate the decoding scheme into multiple parallel branches of weight-shared U-Net decoders, each masked by a single segment. Noise prediction for k-th branch is:

$$\hat{\boldsymbol{\epsilon}}_k = \theta_{dec}(\boldsymbol{h}_{mid}, \boldsymbol{h}_{enc} \odot \boldsymbol{m}_k, t) \tag{2.6}$$

and the output is a sum of region-masked predictions:

$$\hat{\boldsymbol{\epsilon}} = \sum_{k=0}^{K-1} \hat{\boldsymbol{\epsilon}}_k \odot \boldsymbol{m}_k \tag{2.7}$$

2.3.3 Optimization with Denoising Objective

We train our model in an end-to-end manner, driven by the simple DDPM denoising objective. Model weights $\theta = \{\theta_{enc}, \theta_{mid}, \theta_{dec}, \theta_{mask}\}$ are optimized by minimizing the noise prediction

Algorithm 1
Training Masked Diffusion

```
Input: Data $x_0$
Output: Trained model $\theta$
Initialize:   Model weights $\theta$,
  Timesteps T
for iter $= 1$ to Iter$_{total}$ do
  Sample $t \in [1, T]$
  Sample $x_t$ using Eq. 3.1
  Calculate $\hat{\epsilon}$ using Eq. 2.7
  Backprop with Eq. 2.8.
  Update $\theta$.
end for
return $\theta$
```

Algorithm 2
Image and Mask Generation

```
Input: Noise $x_T$, trained model $\theta$
Output:
  Image $\hat{x_0}$ and segmentation $\hat{m}_0$
Initialize: $x_T \sim \mathcal{N}(0, 1)$
for t $= T$ to $1$ do
  Sample $z$ using Eq. 2.10
  Perform reversed diffusion using Eq. 2.9
  if $t = 1$ then
    collect $\hat{m}_0$ using Eq. 2.4
    return $\hat{x_0}$ and $\hat{m}_0$.
  end if
end for
```

loss:

$$L = \mathbb{E}||\epsilon - \hat{\epsilon}||_2^2 \tag{2.8}$$

Unlike previous work, our method does not require a mask regularization loss term [Savarese et al., 2021, Bielski and Favaro, 2019, Chen et al., 2019], which pre-defines mask priors (*e.g.,* object size). Algorithm 1 summarizes training.

2.3.4 Segmentation via Reverse Diffusion

Once trained, we can deploy our model to both segment novel input images as well as synthesize images from noise.

Real Image Segmentation. Given clean input image x_0, we first sample a noisy version x_t through forward diffusion in Eq. 3.1. We then perform one-step denoising by passing x_t to the model. We collect the predicted region masks as the segmentation for x_0 using Eq. 2.4.

Image and Mask Generation. Using reverse diffusion, our model can generate realistic images and their corresponding segmentation masks, starting from a pure noise input $x_T \sim \mathcal{N}(0, 1)$. Reverse diffusion predicts $\boldsymbol{x}_{t-1}$ from $\boldsymbol{x}_t$:

$$\boldsymbol{x}_{t-1} = 1/\sqrt{\alpha_t}(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\theta(\boldsymbol{x}_t, t)) + \sigma_t \boldsymbol{z}, \tag{2.9}$$

$$\boldsymbol{z} \sim \mathcal{N}(0, 1) \quad \text{if} \quad t > 1 \quad \text{else} \quad \boldsymbol{z} = 0. \tag{2.10}$$

where σ_t is empirically set according to the DDPM noise scheduler. We perform T steps of reverse diffusion to generate an image. We also collect its corresponding mask using Eq. 2.9 when $t = 1$. Algorithm 2 summarizes this process.

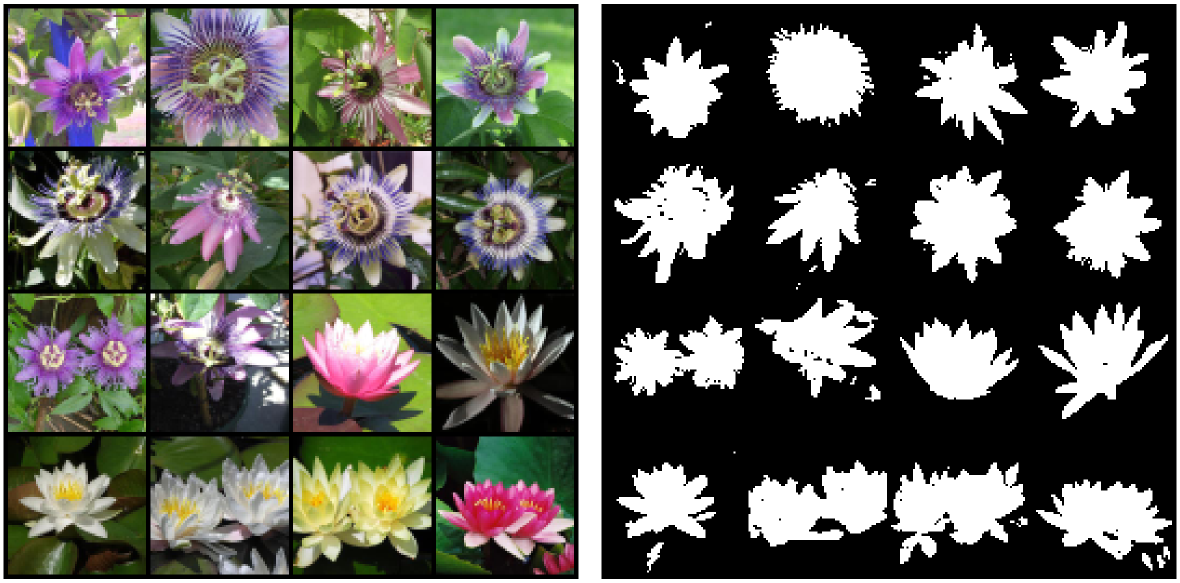
2.4 Experiments

We evaluate on: (1) real image segmentation, (2) image and region mask generation, using Flower [Nilsback and Zisserman, 2008], CUB [Wah et al., 2011], FFHQ [Karras et al., 2019], CelebAMask-HQ [Lee et al., 2020], and ImageNet [Russakovsky et al., 2015].

Evaluation Metrics. For unsupervised segmentation on Flower and CUB, we follow the data splitting in IEM [Savarese et al., 2021] and evaluate predicted mask quality under three commonly used metrics, denoted as Acc., IOU and DICE score [Savarese et al., 2021, Chen et al., 2019]. Acc. is the (per-pixel) mean accuracy of the foreground prediction. IOU is the predicted foreground region's intersection over union (IoU) with the ground-truth foreground region. DICE score is defined as $2\frac{\hat{F} \cap F}{|\hat{F}|}$ [Dice, 1945]. On ImageNet, we evaluate our method on Pixel-ImageNet [Zhang et al., 2020b], which provides human-labeled segmentation masks for 0.485M images covering 946 object classes. We report Acc., IOU and DICE score on a randomly sampled subset, each class containing at most 20 images. For face datasets, we train our model on FFHQ and only report per-pixel accuracy on the CelebAMask test set, using provided ground-truth.

For image and mask generation, we use Fréchet Inception Distance (FID) [Heusel et al., 2017] for generation quality assessment. Since we can not obtain the ground-truth for generated masks, we apply a supervised U-Net segmentation model, pre-trained on respective datasets, to the generated images and measure the consistency between masks in terms of per-pixel accuracy. In addition to quantitative comparisons, we show extensive qualitative results.

Implementation Details. We train Flower, CUB and Face models at both 64×64 and 128×128 resolution. We also train class-conditioned ImageNet models with 64×64 resolution. For all experiments, we use the U-Net [Ronneberger et al., 2015] encoder-middle-decoder architecture similar to [Ho et al., 2020]. We use the decoder architecture as our mask generator and set the number of factorized masks K as 3. For 64×64 the architecture is as follows: The downsampling stack performs four steps of downsampling, each with 3 residual blocks. The upsampling stack is setup as a mirror image of the downsampling stack. From highest to lowest resolution, U-Net stages use $[C, 2C, 3C, 4C]$ channels, respectively. For 128×128 architecture, the down/up sampling block is 5-step with $[C, C, 2C, 3C, 4C]$ channels, each with two residual blocks, respectively. We set $C = 128$ for all models.



(a) **Real Images**

(b) **Segmentation**

Figure 2.3: Segmentation on Flower.

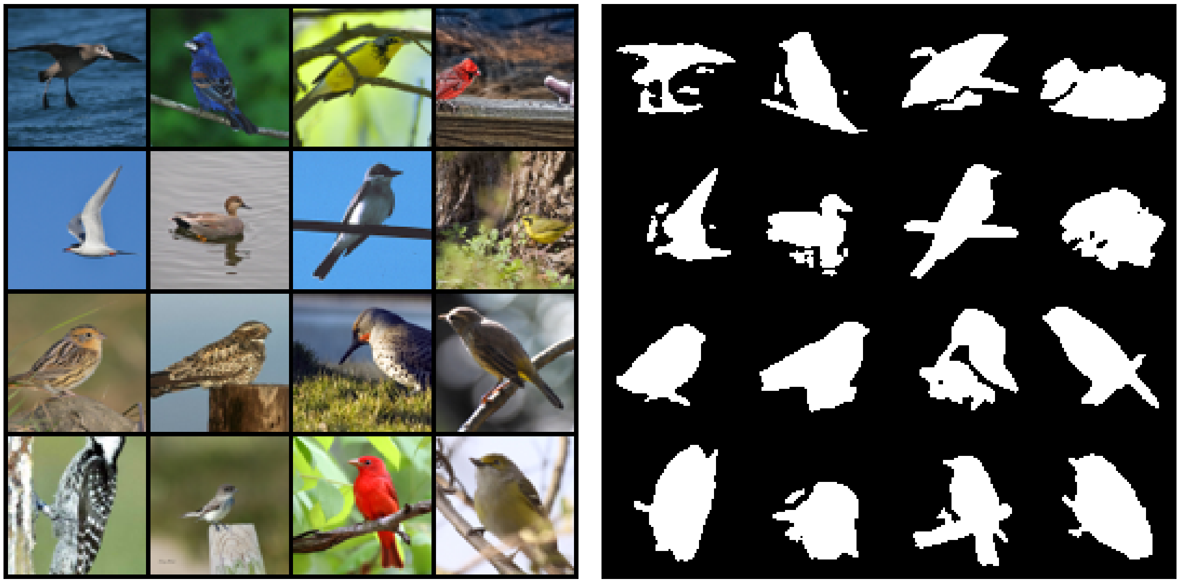
Methods	Acc.	IOU	DICE
GrabCut [Rother et al., 2004]	82.0	69.2	79.1
ReDO [Chen et al., 2019]	87.9	76.4	-
IEM [Savarese et al., 2021]	88.3	76.8	84.6
IEM+SegNet [Savarese et al., 2021]	89.6	78.9	86.0
Ours	**90.1**	**79.7**	**87.2**

Table 2.1: Comparisons on Flower.

We use Adam to train all the models with a learning rate of 10^{-4} and an exponential moving average (EMA) over model parameters with rate 0.9999. For all datasets except ImageNet, we train 64×64 and 128×128 models on 8 and 32 Nvidia V100 32GB GPUS, respectively. For Flower, CUB and FFHQ, we train the models for 50K, 50K, 500K iterations with batch size of 128, respectively. For ImageNet, we train 500K iterations on 32 Nvidia V100 GPUS with batch size 512. We adopt the linear noise scheduler as in Ho *et al.* [Ho et al., 2020] with $T = 1000$ timesteps.

2.4.1 Image Segmentation

To evaluate our method on real image segmentation, we set t as 30 for forward diffusion process. For Flower and CUB, Figures 2.3 and 2.4 show test images and predicted segmentations. Tables 2.1 and 2.2 provide quantitative comparison with representative unsupervised image segmentation methods: GrabCut [Rother et al., 2004], ReDO [Chen et al., 2019] and IEM [Savarese et al., 2021]. As shown in Table 2.1 and Table 2.2, our method outperforms all competitors.



(a) **Real Images**

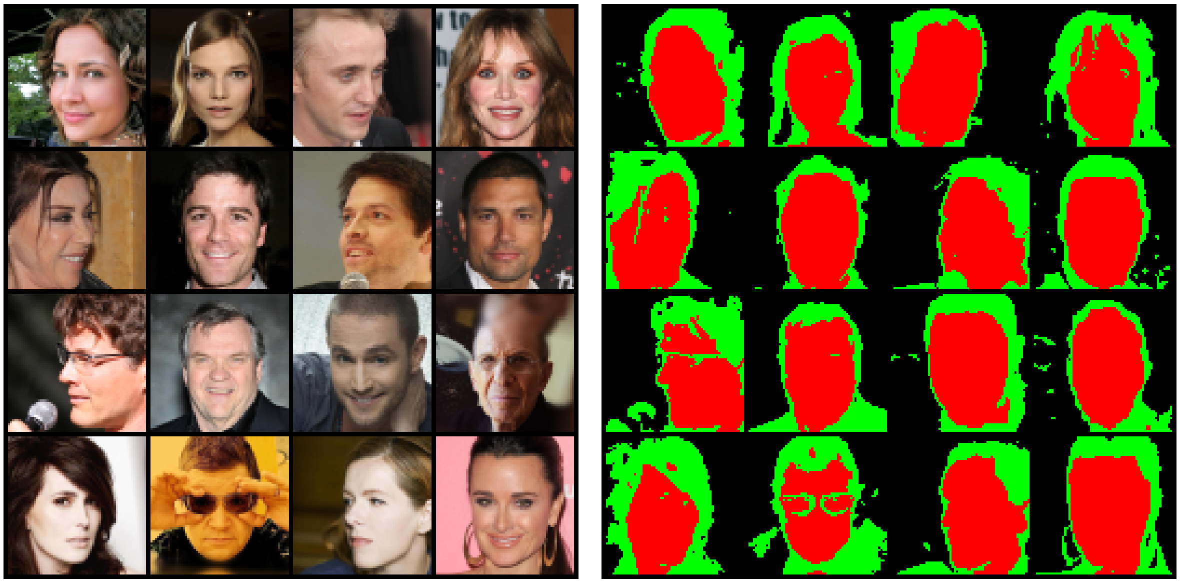
(b) **Segmentation**

Figure 2.4: Segmentation on CUB.

Methods	Acc.	IOU	DICE
GrabCut [Rother et al., 2004]	72.3	36.0	48.7
PerturbGAN [Bielski and Favaro, 2019]	-	38.0	-
ReDO [Chen et al., 2019]	84.5	42.6	-
IEM [Savarese et al., 2021]	88.6	52.2	66.0
IEM+SegNet [Savarese et al., 2021]	89.3	55.1	68.7
Ours	**89.6**	**56.1**	**69.4**

Table 2.2: Comparisons on CUB.

We also visualize the predicted face parsing results on FFHQ and CelebAMask datasets in Figure 3.2(c)(d) and Figure 2.5. Our model learns to accurately predict three segments corresponding to semantic components: skin, hair, and background. Note that the term "hair" refers to the structure surrounding the face, as the hair component is predominant in these regions. This particular semantic partitioning emerges from our unsupervised learning objective, without any additional prior. With ground-truth provided on CelebAMask-HQ, we also compare the pixel accuracy and mean of IOU with a supervised U-Net and DatasetDDPM [Baranchuk et al., 2022]. For the former, we train a supervised segmentation model with 3-class cross-entropy loss. For the unsupervised setting, we perform K-means (K=3) on the pre-trained DDPM, denoted as DatasetDDPM-unsup. Table 2.3 shows that we outperform DatasetDDPM by a large margin and achieve a relatively small performance gap with a supervised U-Net.



(a) **Real Images**

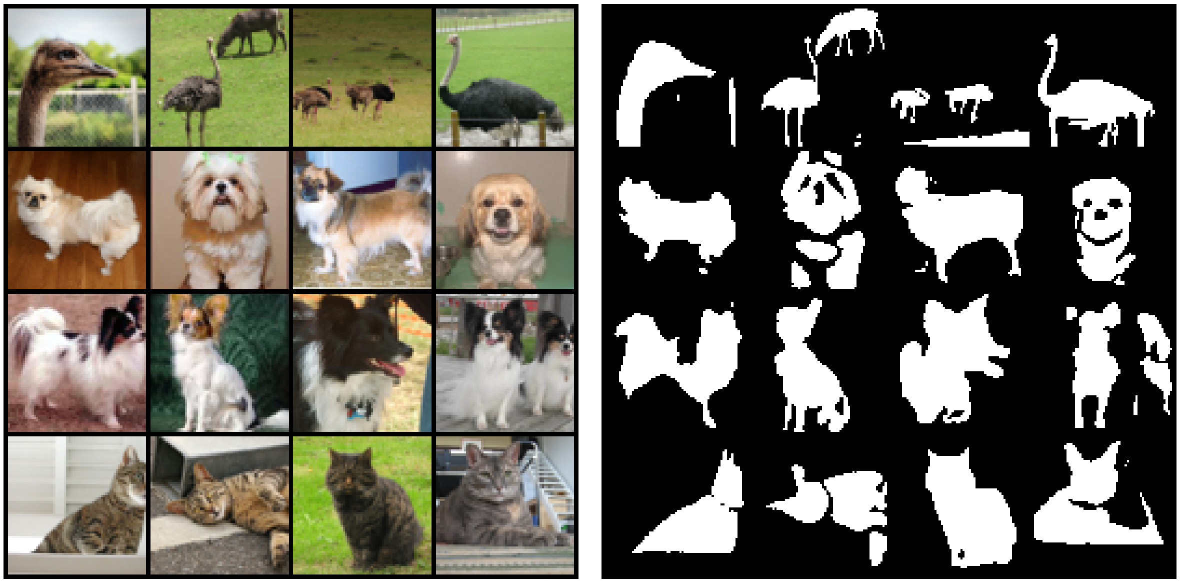
(b) **Segmentation**

Figure 2.5: Segmentation on CelebA.

Methods	Acc.	mIOU
Supervised UNet	95.7	90.2
DatasetDDPM-unsup. [Baranchuk et al., 2022]	78.5	69.3
Ours	87.9	80.3

Table 2.3: Segmentation comparisons on CelebA.

Figure 2.6 shows the accurate segmentation results for ImageNet classes: ostrich, pekinese, papillon, and tabby. We compare with supervised U-Net and DatasetDDPM-unsup in Table 2.4. We show more visualizations in Appendix Section 2.5.3.



(a) **Real Images**

(b) **Segmentation**

Figure 2.6: Segmentation on ImageNet.

Methods	Acc.	mIOU
Supervised UNet	85.7	74.1
DatasetDDPM-unsup. [Baranchuk et al., 2022]	74.1	60.4
Ours	80.7	67.7

Table 2.4: Segmentation comparisons on ImageNet.

2.4.2 Image and Mask Generation

We evaluate our method on image and mask generation. As shown in Figure 2.7, 2.8, 3.2(c)(d) and 2.11, our method is able to generate realistic images. In the upper row of Table 2.5, we see a consistent quality improvement over the original DDPM. This suggests our method as a better architecture than standard U-Net through separating computational power to each individual image segment, which may benefit the denoising task during training. More importantly, our method can produce accurate corresponding masks, closely aligned with the semantic partitions in the generated image.

We evaluate the segmentation quality. Since there is no ground-truth mask provided for generated images, we apply the U-Net segmentation models (pre-trained on respective labeled training sets) to the generated images to produce reference masks. We measure the consistency between the reference and the predicted parsing results in terms of pixel-wise accuracy. We compare our method with a pre-trained DDPM baseline, in which we first perform image generation, then pass them to DatasetDDPM-unsup to get masks. As shown in Table 2.5 (bottom), our method consistently achieves better segmentation on generated images than the DDPM baseline. Note that, different from the two-stage baseline, our method performs the computation in a single stage, generating image and mask simultaneously. Appendix Section 2.5.3 shows more visualizations.

Table 2.5: Image and mask generation comparison on all datasets. (upper: FID(↓) bottom: Acc. (↑))

Models	Flower-64	Flower-128	CUB-64	CUB-128	FFHQ-64	FFHQ-128	ImageNet-64
DDPM	15.81	14.62	14.45	14.01	13.72	13.35	7.02
Ours	**13.33**	**11.50**	**10.91**	**10.28**	**12.02**	**10.79**	**6.54**
DDPM	80.5	82.9	84.2	83.7	84.2	84.2	71.2
Ours	**92.3**	**92.7**	**91.4**	**91.2**	**90.3**	**90.7**	**84.1**

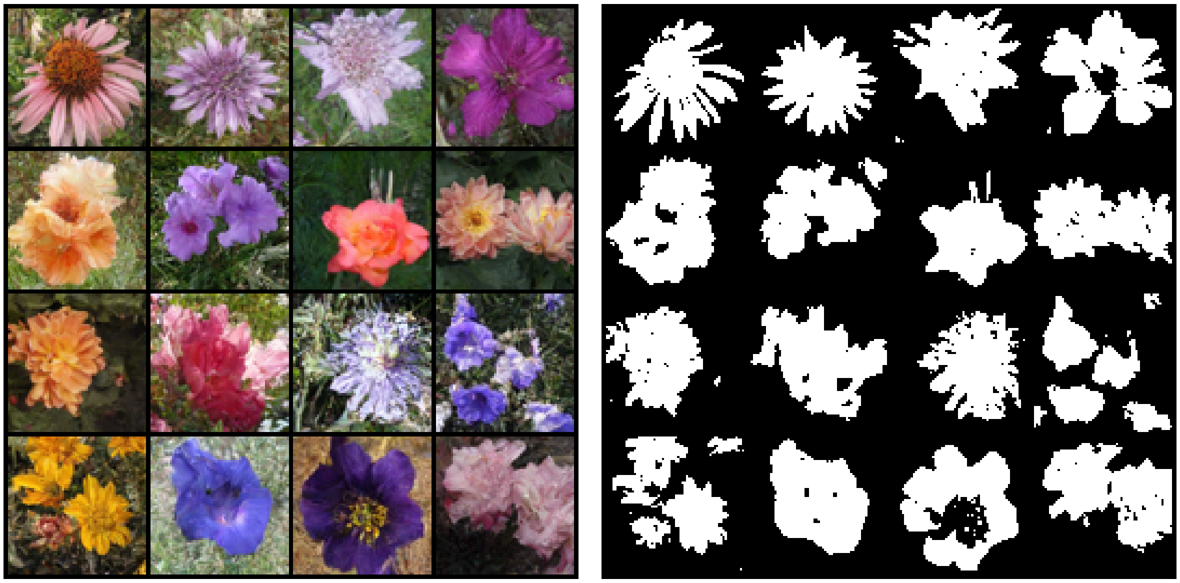
2.4.3 Ablation Study and Analysis

Multi-branch Decoders with Weight Sharing. Separating computation in multi-branch decoders with weight sharing is an essential design in our method. We show the effectiveness of this design by varying how to apply factorized masks in our decoding scheme: (1) concat: we use single branch to take concatenation of $\boldsymbol{h}$ and $\boldsymbol{m}$. (2) masking $\boldsymbol{h}_{mid}$: we use $\boldsymbol{m}$ to mask $\boldsymbol{h}_{mid}$ instead of $\boldsymbol{h}_{enc}$. (3) w/o weight sharing: we train decoders separately in our design. Table 2.6 shows separate design consistently yields better visual features than other designs for CUB. This suggests that our design benefits from fully utilizing mask information in the end-to-end denoising task and avoids a trivial solution where masks are simply ignored. **Investigation on Mask Factorization.** Our architecture is able to generate factorized representations, each representing a particular segment of the input image. We show this by visualizing the individual channels from softmax layer output in our mask generator. As shown in Figure 2.9, skin, hair, and background are separated in different channels.

Mask Refinement along Diffusion Process. In the DDPM Markov process, the model implicitly formulates a mapping between noise and data distributions. We validate that this occurs for both images and latent region masks by visualizing image and mask generation along the sequential reversed diffusion process in Figure 2.10. We observe gradual refinement as denoising steps approach $t = 0$.

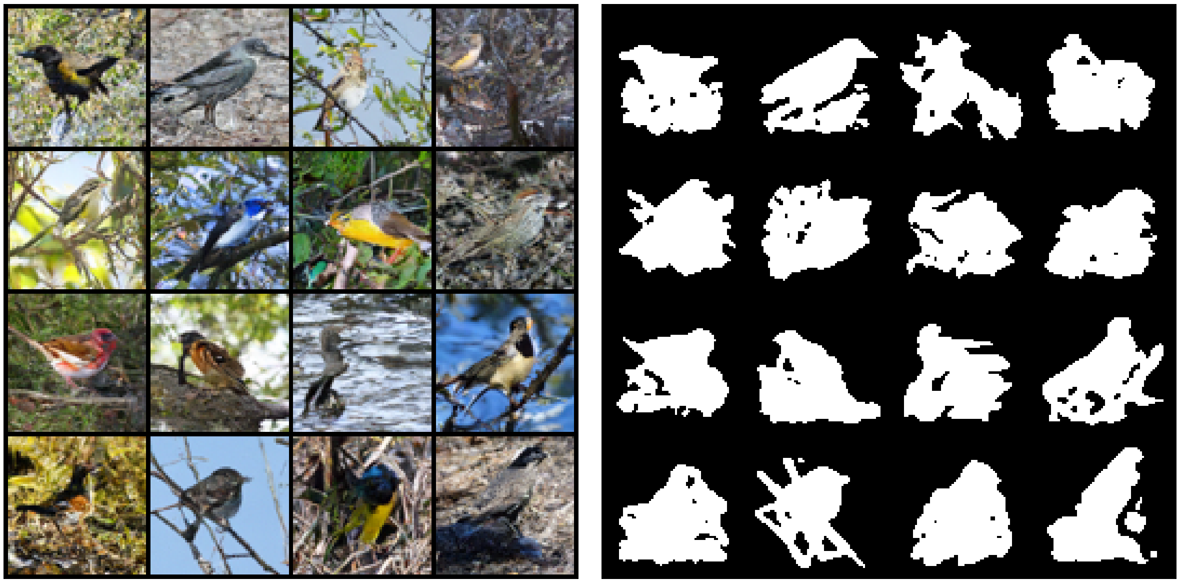
Face Interpolation. We also investigate the face interpolation task on FFHQ. Similar to standard DDPM [Ho et al., 2020], we perform the interpolation in the denoising latent space with 250 timesteps of diffusion. Figure 2.12 shows good reconstruction in both pixels and region masks, yielding smoothly varying interpolations across face attributes such as pose, skin, hair, expression, and background.

Zero-shot Object Segmentation. We evaluate zero-shot object segmentation on both PASCAL VOC 2012 [Everingham et al.] and DAVIS-2017 videos [Pont-Tuset et al., 2017]. Baseline DDPM generation is not solved for these datasets when training from scratch without external large-scale datasets (*e.g.,* LAION [Schuhmann et al., 2022] used in Stable Diffusion [Rombach et al., 2022]). We directly adopt zero-shot transfer from our pre-trained ImageNet model by applying the conditional label mapping. We detail the mapping rule in Appendix Section 2.5.4. Figure 2.13 shows the accurate segmentation results for images of classes: aeroplane, monitor, person, and sofa from VOC. Since our method does not require any pixel labels, we evaluate the performance of each object class individually. Our method achieves a favorable high accuracy of **0.78** and mIOU of **0.54** when averaging over all 20 classes. We also report the results for each individual class in Appendix Section 2.5.5. We also show video segmentation on DAVIS-2017 in Figure 2.14 and Appendix Section 2.5.6, without any labeled video pre-training.



(a) **Generated Images** (b) **Generated Masks**

Figure 2.7: Generation on Flower.



(a) Generated Images. (b) Generated Masks.

Figure 2.8: Generation on CUB.

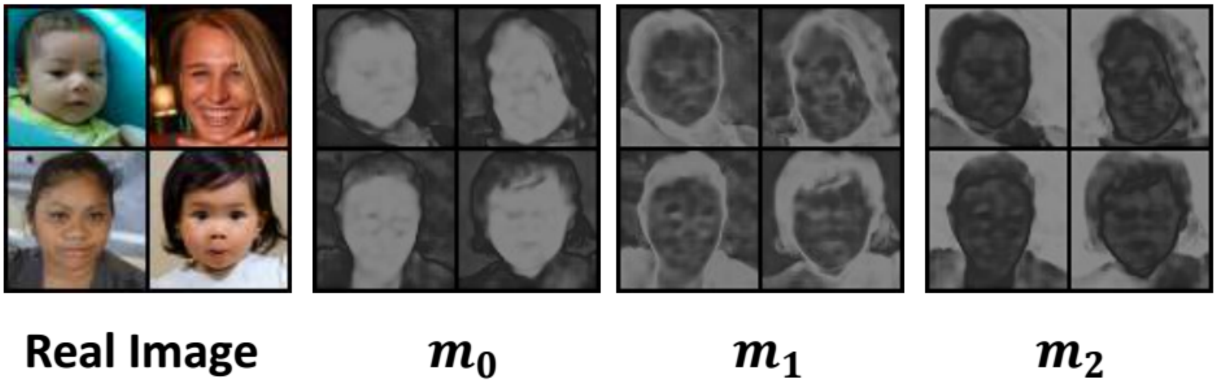


Figure 2.9: Mask factorization (3 parts) on FFHQ.

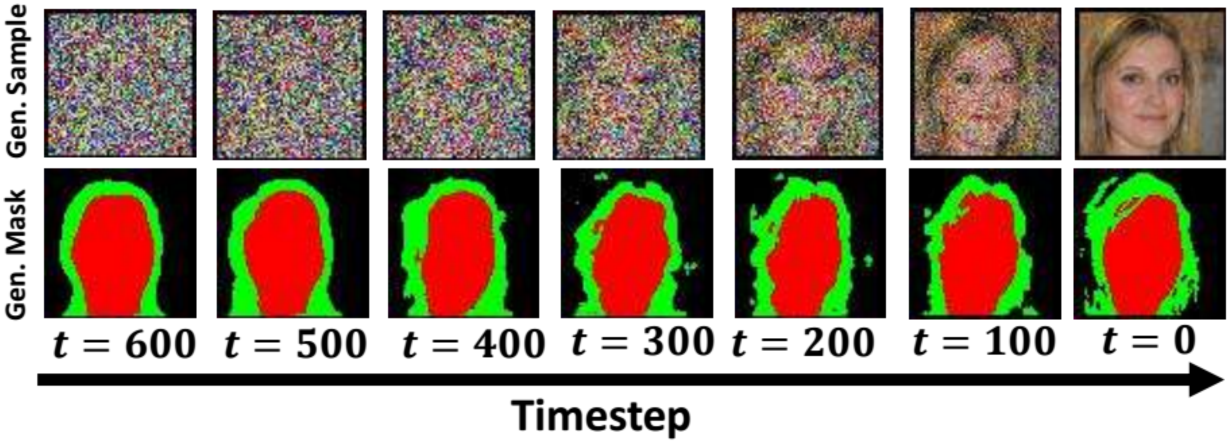
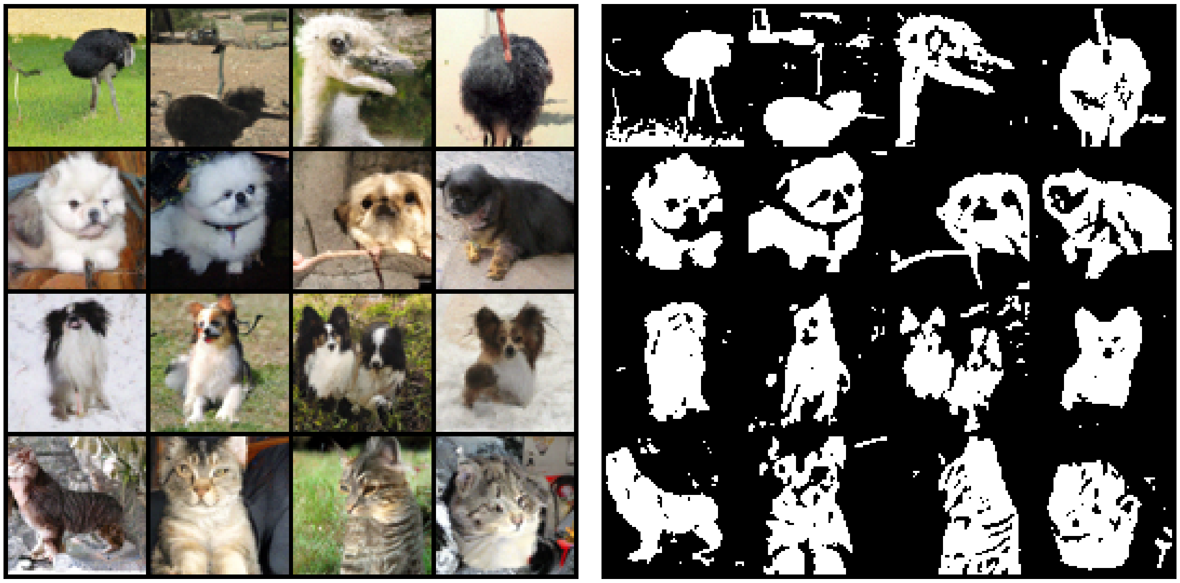


Figure 2.10: Generation refinement along diffusion.



(a) **Generated Images** (b) **Generated Masks**

Figure 2.11: Conditional generation on ImageNet.

Methods	IOU.(↑)	FID (↓)
Concat	20.7	14.21
Masking $\boldsymbol{h}_{mid}$	20.2	14.33
w/o weight sharing	50.5	17.21
Ours	**56.1**	**10.28**

Table 2.6: Ablations of decoding scheme on CUB.

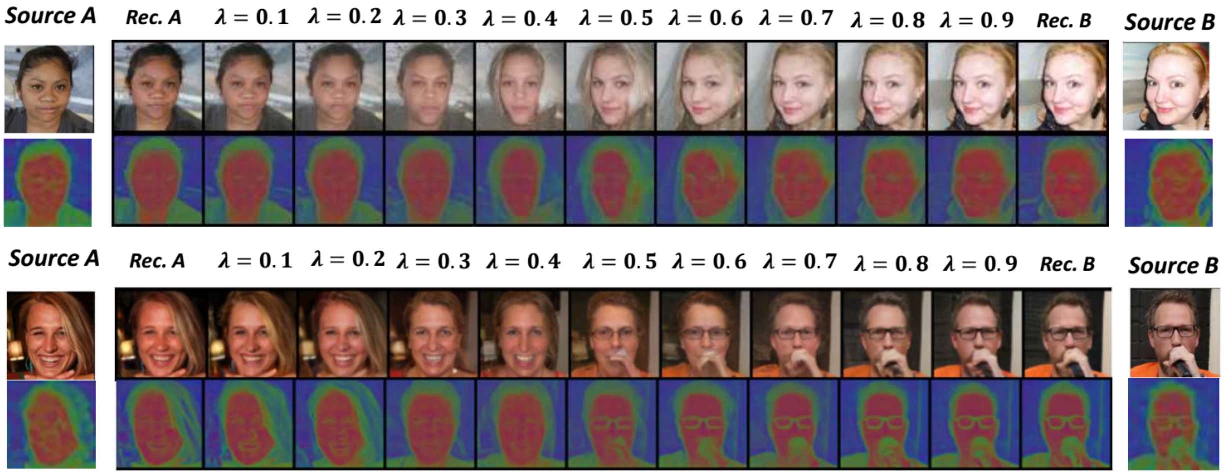
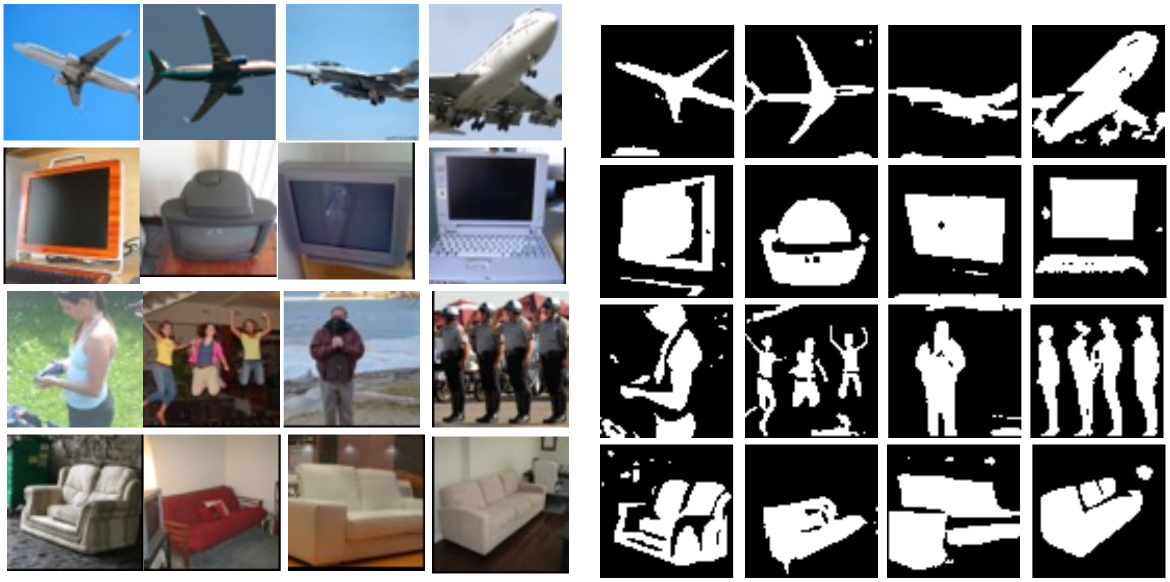
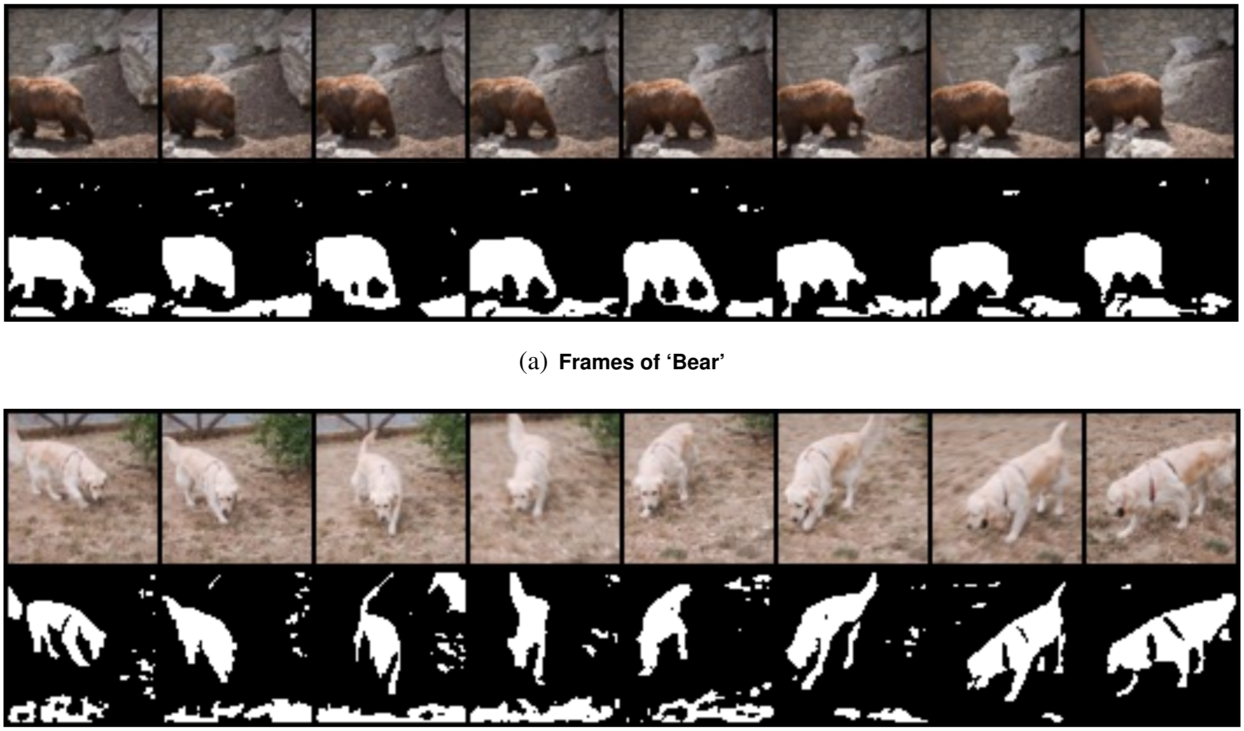


Figure 2.12: Interpolations of FFHQ with 250 timesteps of diffusion.



(a) **Real Images** (b) **Segmentation**

Figure 2.13: Segmentation on VOC-2012.



(a) **Frames of ‘Bear’**

(b) **Frames of ‘Dog’**

Figure 2.14: Segmentation on DAVIS-2017.

2.5 Appendix

2.5.1 Hierarchical Factorized Diffusion

We conduct a further investigation is to reorganize our architectural design to support hierarchical mask factorization in place of a flat set of K regions. We formulate a hierarchical factorized diffusion architecture to progressively refine segmentation results from a coarse initial prediction to a fine, detailed final segmentation. This approach helps in capturing both global context and fine details in the segmentation task. As shown in Figure 2.15, the first level replicates the factorized diffusion architecture depicted in Figure 3.1 to generate initial region masks of $m_0^0, m_1^0, ...$, each applied on the noisy input for the next level factorized diffusion process. Each branch of the second level architectures generates finer representations of region masks $m_0^1, m_1^1, ...$, constructing the final denoising output as $\sum_i m_i^0 \frac{(h_i^0 + \sum_j h_j^1 m_j^1)}{2}$. The nested architectural design can be instantiated as infinite levels of factorized diffusion, which is a promising way to handle multi-scale scenes. AS a proof of concept, we conduct the experiment on the shape 3D dataset [Burgess and Kim, 2018] with a 2-level hierarchy. We first visualize each level's region mask in Figure 2.16. We observe that for the first level generates a coarse-level segmentation, based on which, second-level factorized diffusion generates fine-level segmentations of 3d shapes. Figure 2.17 provides a more direct visualization of partitions at each level through a 3-class mapping.

2.5.2 Additional Segmentation Results

We show more segmentation results for Flower, CUB, FFHQ, CelebA and ImageNet. As shown in Figures 2.19, 2.20, 2.21, 2.22, and 2.23, our method consistently predicts accurate segmentations for real image inputs.

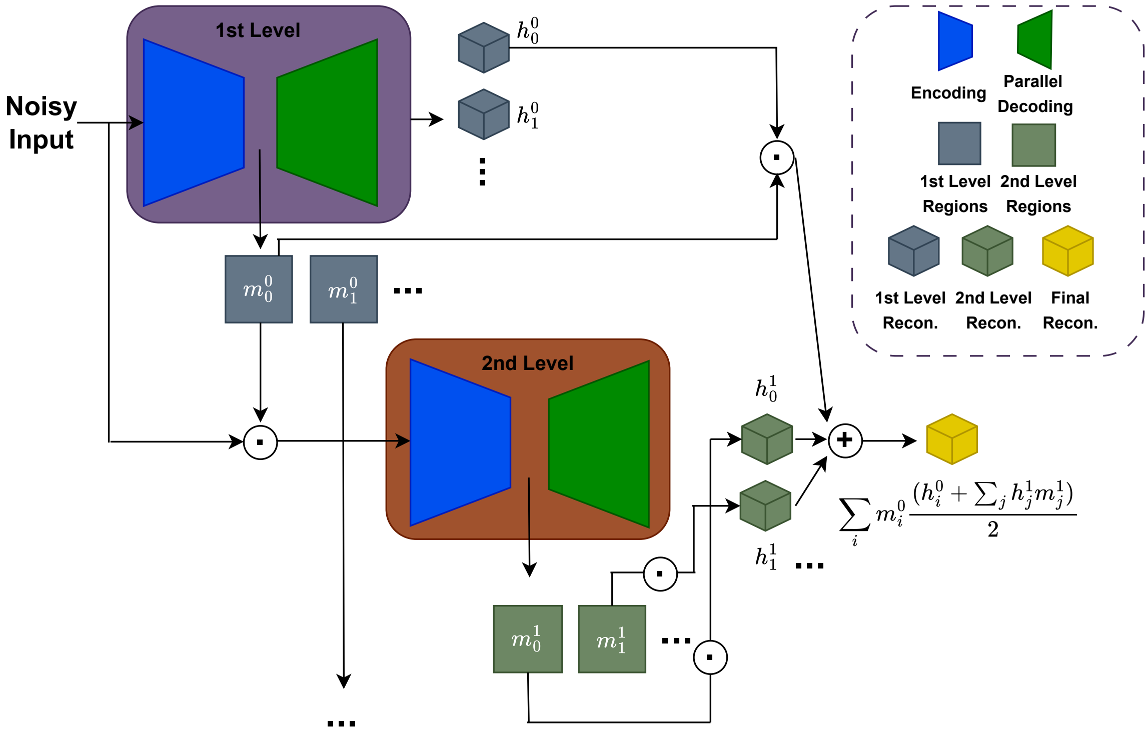


Figure 2.15: **Hierarchical factorized diffusion architecture.**

2.5.3 Additional Generation Results

We show more generation results for Flower, CUB, FFHQ, and ImageNet (classes: flamingo, water buffalo, garbage truck, and sports car). As shown in Figures 2.24, 2.25, 2.26, and 2.27, our method consistently produces images and masks with high quality.

2.5.4 Label Mapping for Zero-shot Transfer

At the current stage of diffusion model research, generation is not solved for PASCAL VOC when training from scratch without an extrernal large-scale dataset (*e.g.,* LAION used in stable diffusion). There is no baseline DDPM model that can achieve this. As such, we adopt our conditional ImageNet model to perform zero-shot segmentation on VOC by mapping class labels from ImageNet to VOC.

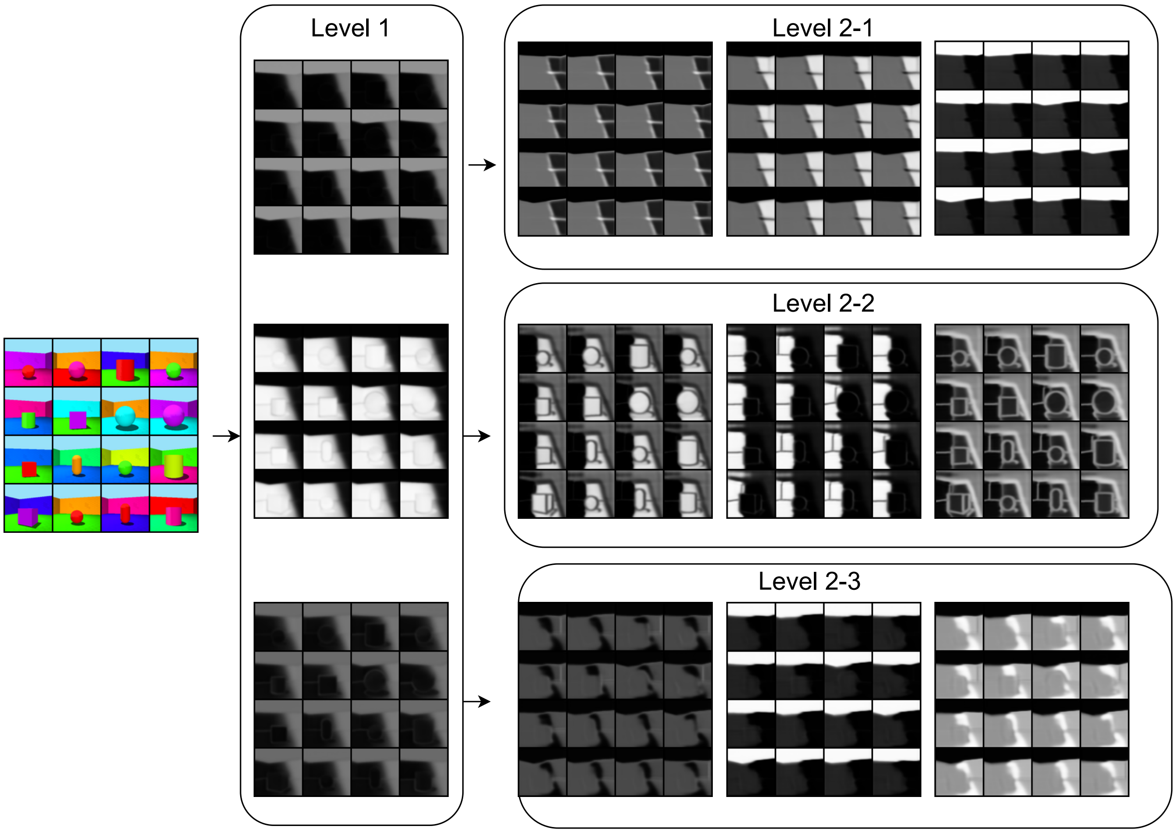


Figure 2.16: **Mask factorization for each level.** ***Level 1:*** visualization of each mask channel at the first level. ***Level 2-1, 2-2, 2-3:*** visualization of each mask channel per branch at the second level.

2.5.5 Additional Zero-shot Results on VOC

We report pixel accuracy and mIOU of each class in VOC in Table **??**, which demonstrates that our method can achieve reasonable high performance. We also provide more segmentation results of 'bicycle', 'chair', 'potted plant' and 'train' in Figure 2.28.

2.5.6 Additional Zero-shot Results on DAVIS

We provide more DAVIS-2017 video segmentation results of 'classic-car', 'dance-jump' in Figure 2.31.

Table 2.7: We perform class label mapping from ImageNet to VOC, and report zero-shot transfer Accuracy and mIOU per class on VOC validation dataset.

VOC Class.	ImageNet Class.	Num. of VOC-val Image	Accuracy	mIOU
1:aeroplane	895:warplane	136	0.82	0.57
2:bicycle	671:mountain-bike	108	0.79	0.47
3:bird	94:hummingbird	168	0.83	0.58
4:boat	814:speedboat	115	0.81	0.51
5:bottle	907:wine-bottle	133	0.76	0.47
6:bus	779:school-bus	114	0.73	0.54
7:car	817:sports-car	191	0.74	0.48
8:cat	281:tabby	206	0.82	0.66
9:chair	765:rocking-chair	175	0.75	0.64
10:cow	346:water-buffalo	102	0.82	0.45
11:diningtable	532:dining-table	89	0.69	0.62
12:dog	153:maltese-dog	204	0.82	0.67
13:horse	603:horsecart	104	0.84	0.53
14:motorbike	670:motorscooter	117	0.76	0.52
15:person	981:ballplayer	584	0.77	0.46
16:potted plant	883:vase	116	0.74	0.46
17:sheep	348:ram	89	0.84	0.64
18:sofa	831:studio-couch	109	0.73	0.51
19:train	466:bullet-train	126	0.76	0.56
20:tv/monitor	664:monitor	106	0.73	0.47
Average	-	-	0.78	0.54

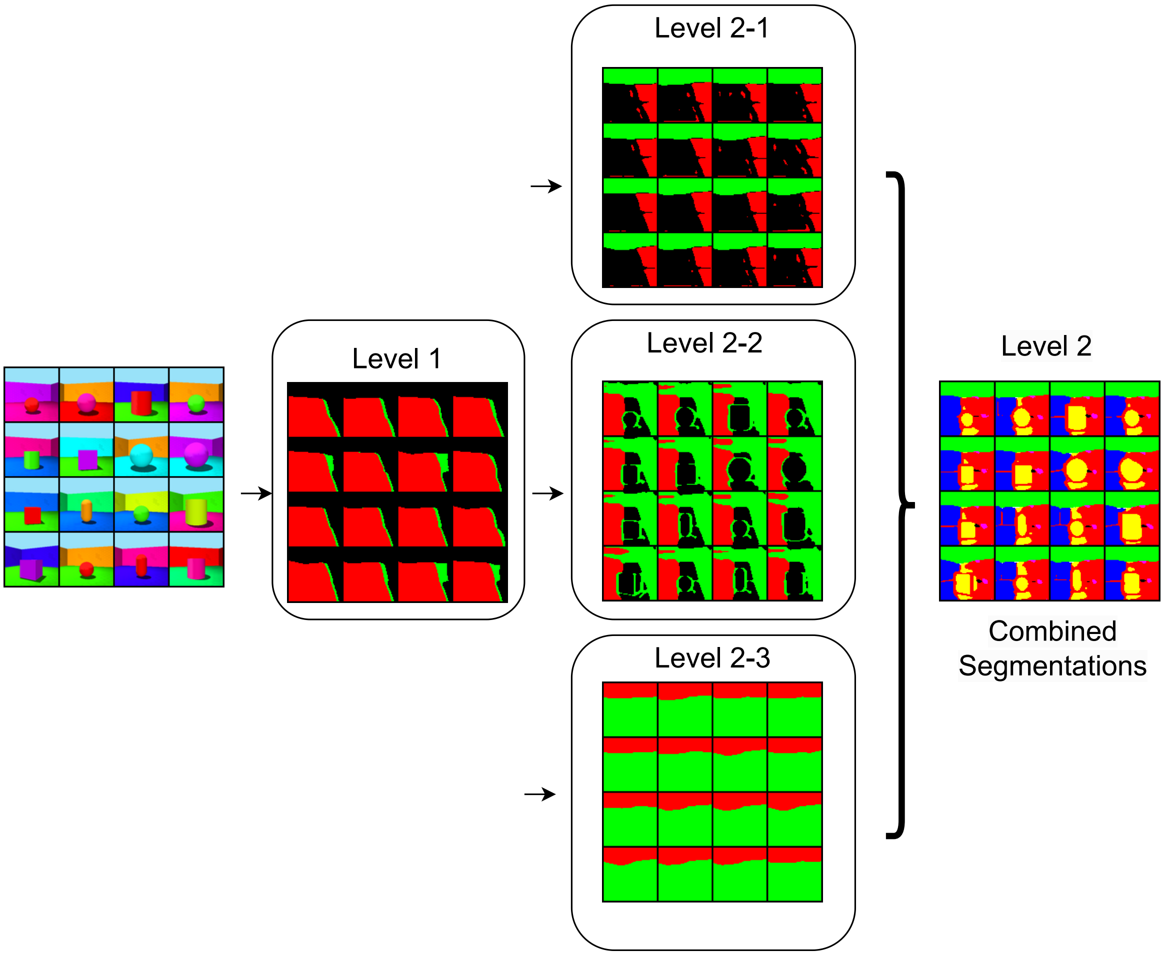
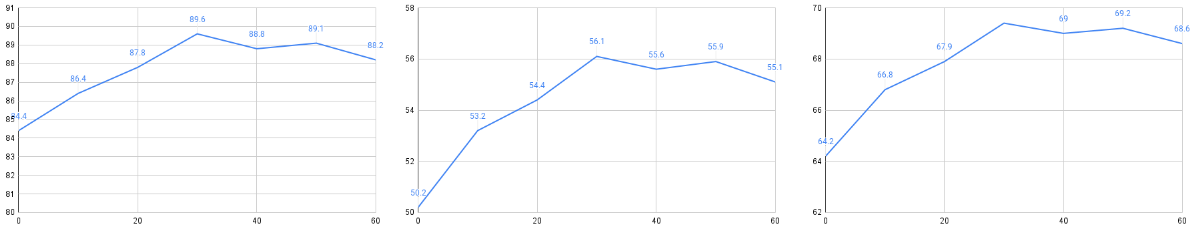


Figure 2.17: **Segmentations for each level.** ***Level 1:*** 3-color-coded region assignments at the first level. ***Level 2-1, 2-2, 2-3:*** 3-color-coded region assignments per branch at the second level. ***Level 2 combined segmentations:*** 9-color-coded region assignments at the second level.

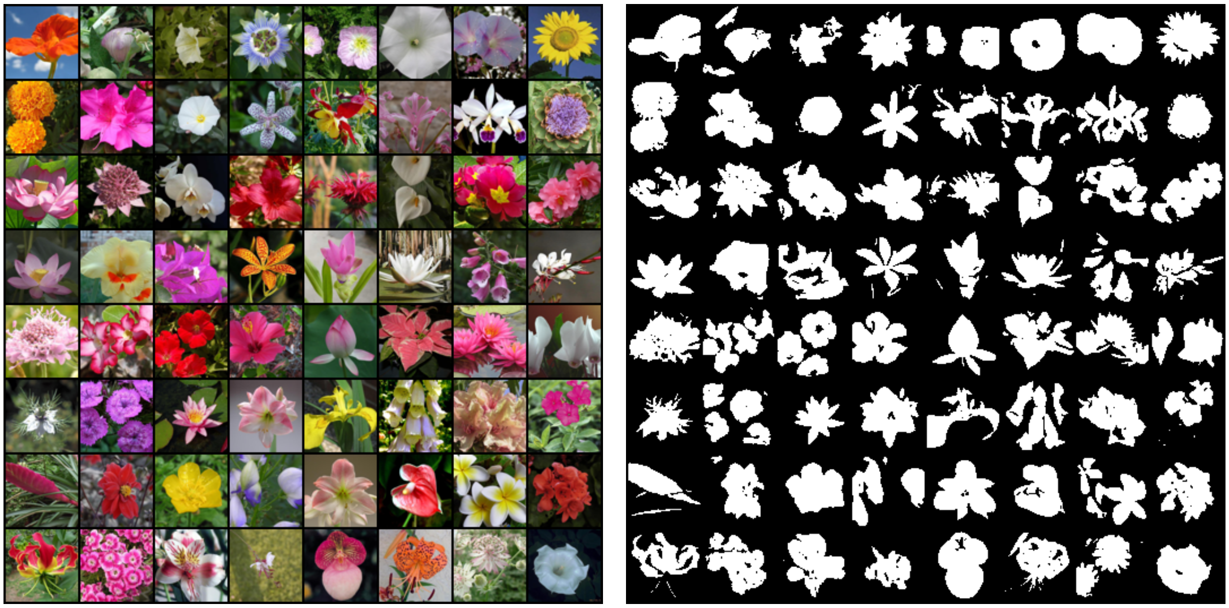


(a) **Acc.**

(b) **IOU.**

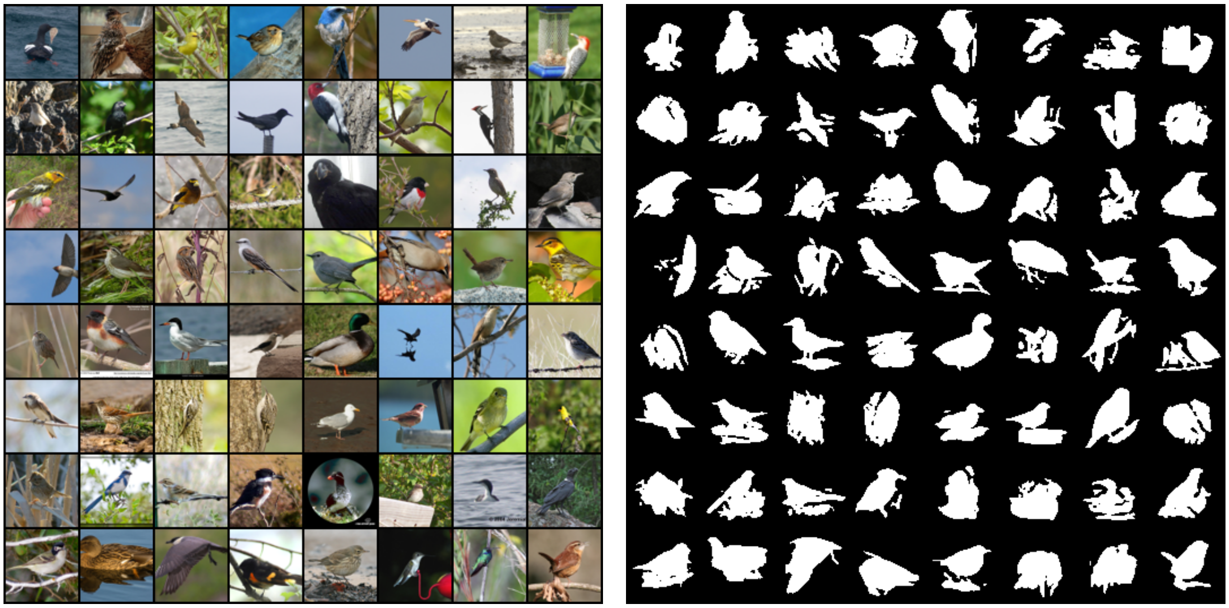
(c) **DICE**

Figure 2.18: **Segmentation results on CUB with** $t \in \{0, 10, 20, 30, 40, 50, 60\}$**.**



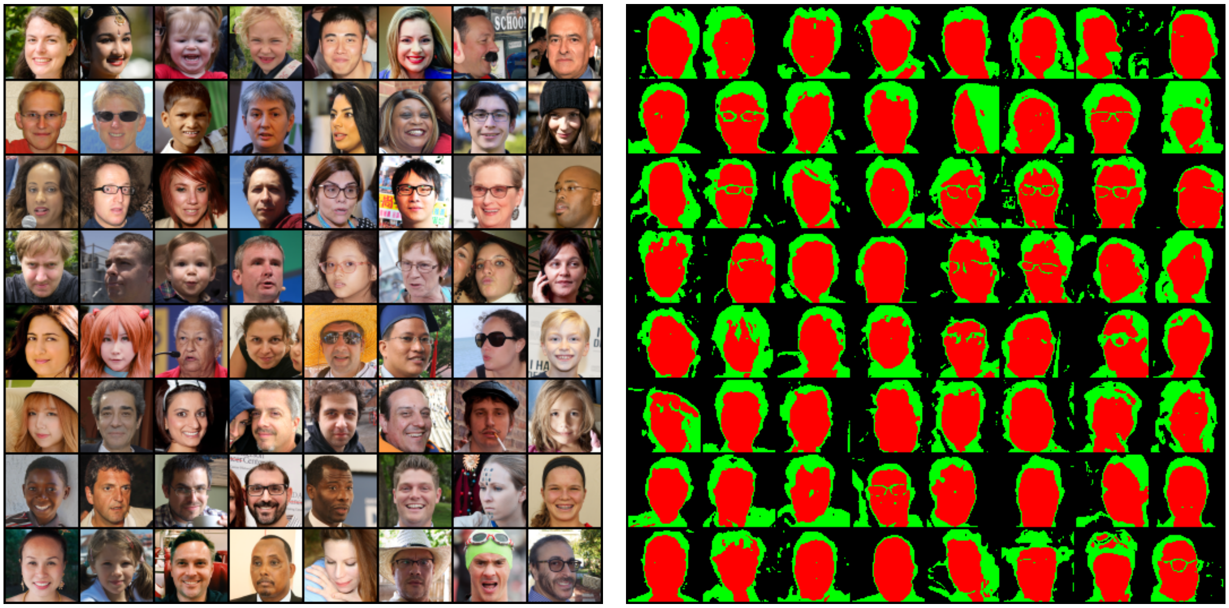
(a) **Real Images** (b) **Segmentation**

Figure 2.19: Segmentation on Flower.



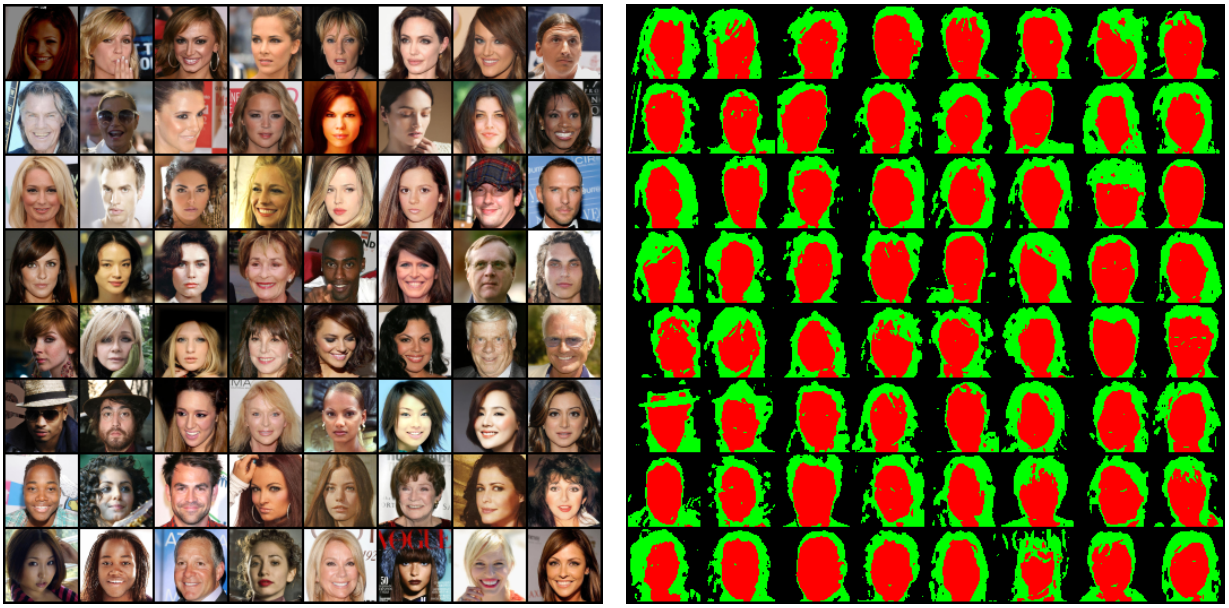
(a) **Real Images** (b) **Segmentation**

Figure 2.20: Segmentation on CUB.



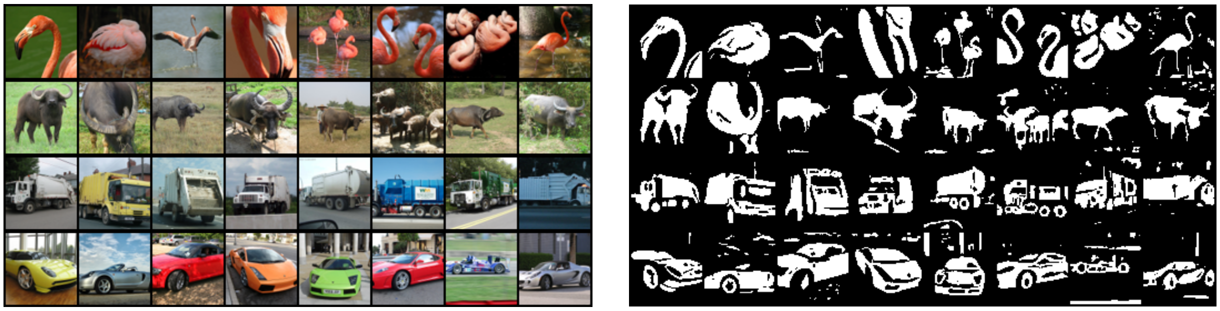
(a) **Real Images** (b) **Segmentation**

Figure 2.21: Segmentation on FFHQ.



(a) **Real Images** (b) **Segmentation**

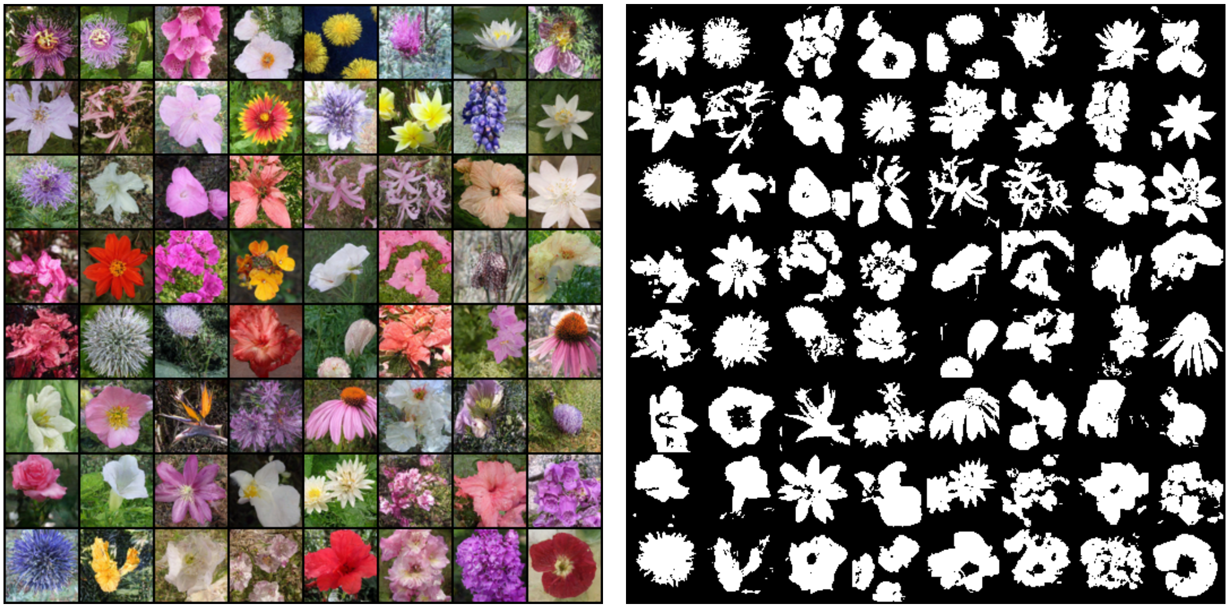
Figure 2.22: Segmentation on CelebA.



(a) **Real Images**

(b) **Segmentation**

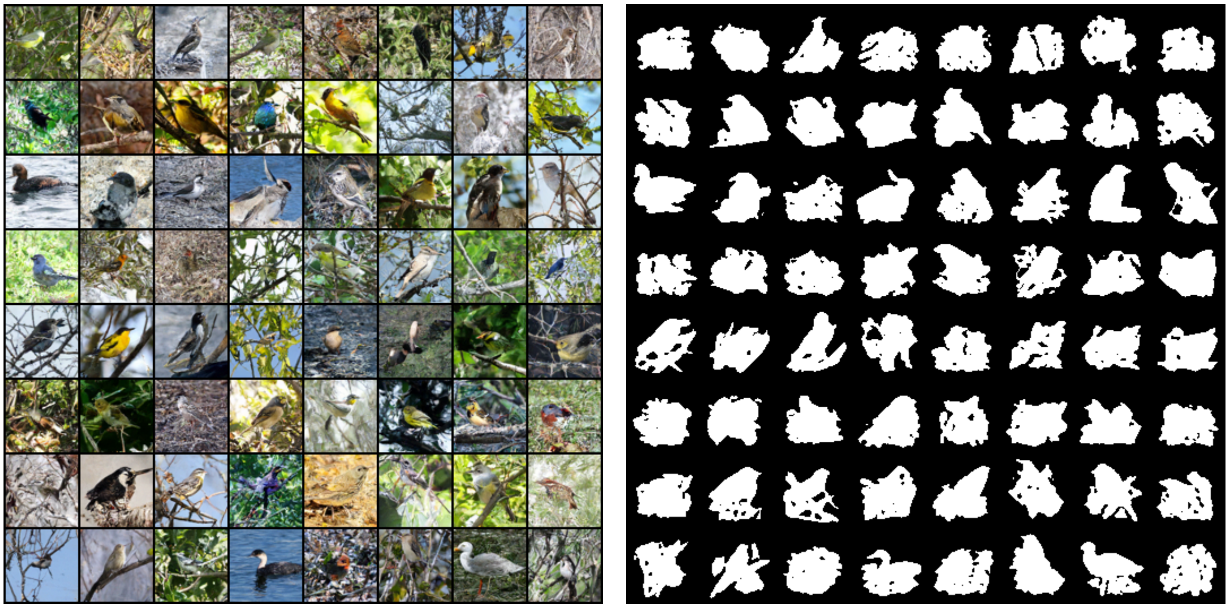
Figure 2.23: Segmentation on ImageNet.



(a) **Generated Images**

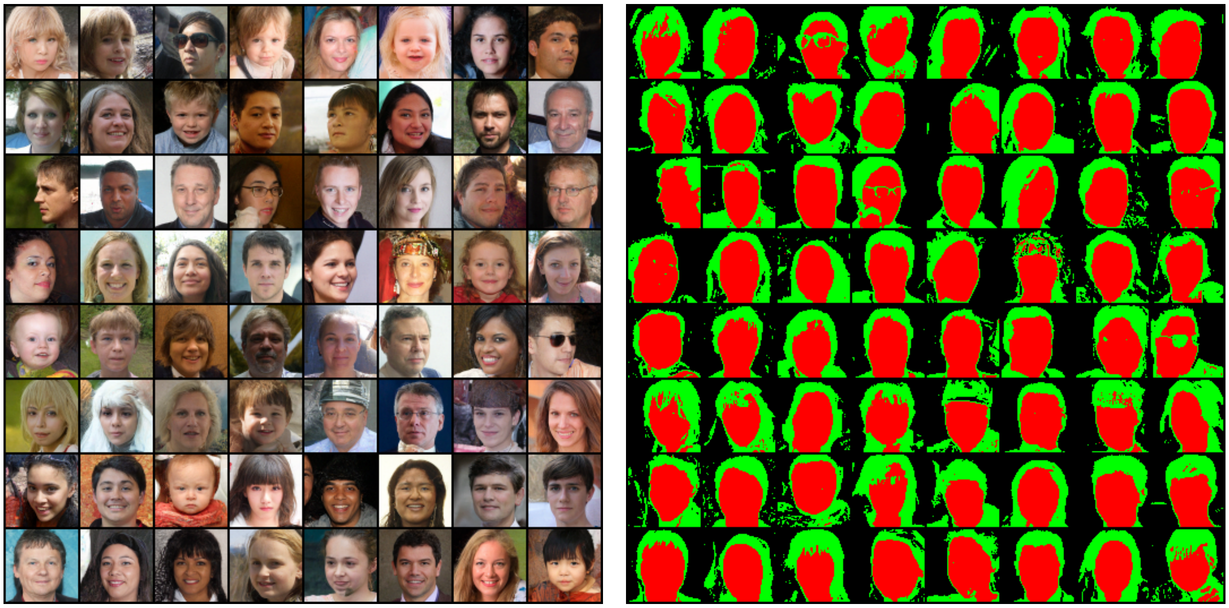
(b) **Generated Masks**

Figure 2.24: Generation on Flower.



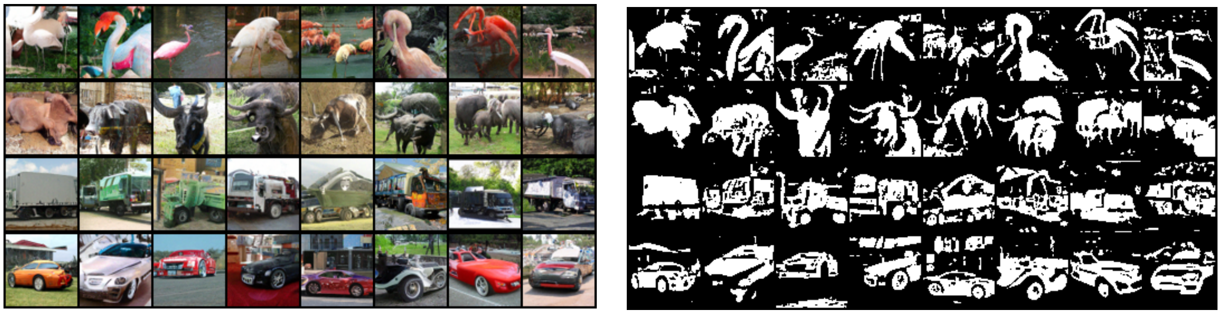
(a) **Generated Images** (b) **Generated Masks**

Figure 2.25: Generation on CUB.



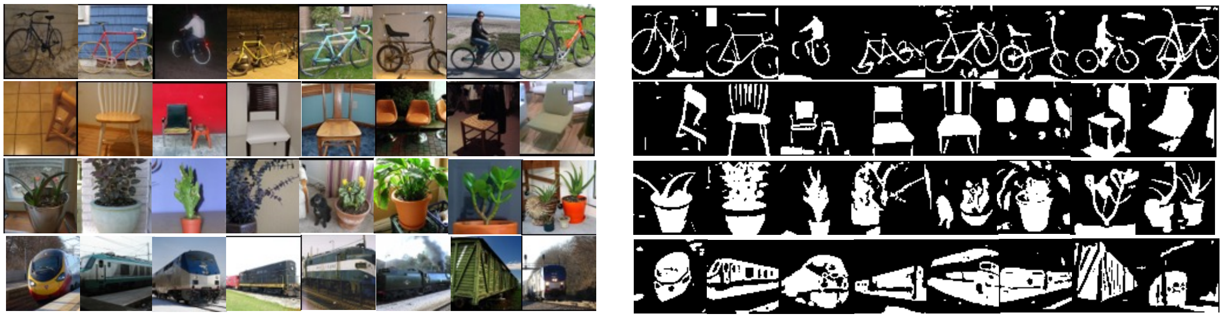
(a) **Generated Images** (b) **Generated Masks**

Figure 2.26: Generation on FFHQ.



(a) **Generated Images** (b) **Generated Masks**

Figure 2.27: Conditional ImageNet generation.



(a) **Real Images** (b) **Segmentation**

Figure 2.28: Segmentation on VOC-2012.

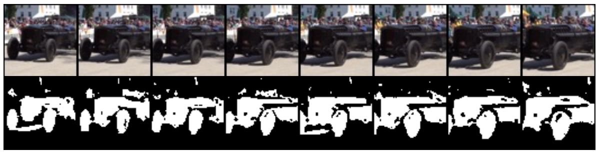


Figure 2.29: **Frames of 'Classic-car'**

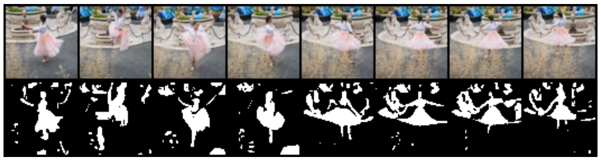


Figure 2.30: **Frames of 'Dance-jump'**

Figure 2.31: Segmentation on DAVIS-2017.

2.6 Summary

We propose a factorized architecture for diffusion models that is able to perform unsupervised image segmentation and generation simultaneously, while being trained once, from scratch, for image generation via denoising alone. Using model architecture as a constraint, via carefully designed component factorization and parallel decoding schemes, our method effectively and efficiently bridges these two challenging tasks in a unified framework, without the need of fine-tuning or alternating the original DDPM training objective. Our work is the first example of engineering an architectural bottleneck so that learning a desired end task becomes a necessary byproduct of training to generate.

Our work is at the stage of a new architectural design for diffusion-based segmentation and generation, with 2- or 3-class segmentation results demonstrating improvements across multiple datasets, scaling up to ImageNet. Our initial investigation into hierarchical extensions suggests a promising future path towards handling complex scenes.

CHAPTER 3
GENERATIVE LIFTING OF MULTIVIEW TO 3D FROM UNKNOWN POSE: WRAPPING NERF INSIDE DIFFUSION

3.1 Introduction

Structure from motion is a well-studied problem in computer vision, with a substantial history of research focusing on the specific task of reconstructing a 3D scene from a collection of 2D images captured from different viewpoints. When the 3D pose (camera extrinsics) for each 2D view is unknown, classic approaches [Snavely et al., 2006, Agarwal et al., 2011] explicitly estimate correspondence between 2D views (*e.g.,* by matching local feature descriptors) prior to optimizing a shared 3D geometry whose reprojections are consistent with those views. Neural Radiance Fields (NeRFs) [Mildenhall et al., 2020, Barron et al., 2021, Martin-Brualla et al., 2021, Barron et al., 2023] have led a revolution toward widespread use of differentiable 3D scene representations [Mildenhall et al., 2020, Fridovich-Keil et al., 2022, Kerbl et al., 2023] that are compatible with deep learning techniques. However, the problem of jointly solving for both the 3D reconstruction and the pose, when neither is known a priori, remains an open problem. Recent attempts to connect learning of camera pose with NeRFs operate under simplifying assumptions, such as coarse pose initialization (only learning adjustments) [Lin et al., 2021] or front-facing (as opposed to arbitrary 360°) views of the scene [Wang et al., 2021].

In parallel with the development of differentiable 3D representations, progress across a variety of paradigms for generative models [Goodfellow et al., 2014, Kingma and Welling, 2014, Ho et al., 2020], has transformed the landscape for designing and training systems using deep learning. Learning to synthesize data provides an unsupervised training objective and scaling compute, parameters, and datasets is a path toward foundation models [Bommasani et al., 2021] whose feature representations can subsequently be repurposed to specific downstream tasks. However, large-scale foundation models are not the only setting in which generative learning is appropriate. Nor is

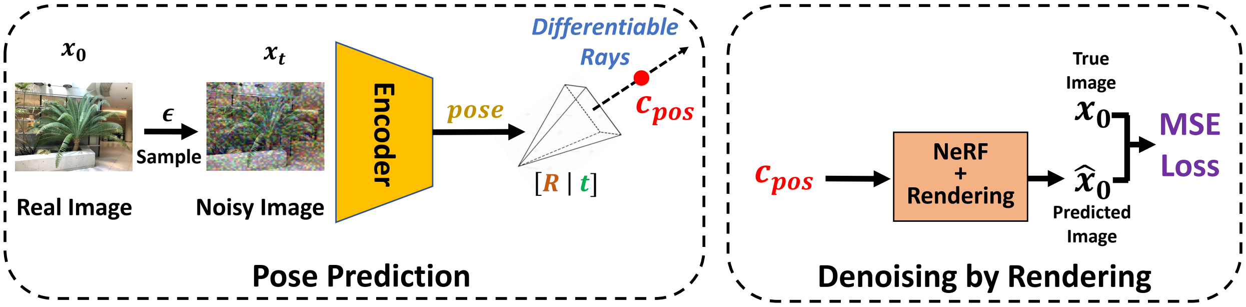


Figure 3.1: **Wrapping NeRF inside Diffusion.** We learn a 3D scene reconstruction by training a denoising diffusion model (DDPM) on a dataset of 2D views of the scene. The architecture of our DDPM consists of two components. ***Left:*** An *Encoder* predicts the pose of a single noisy 2D input image. ***Right:*** A *NeRF* is rendered from the predicted camera pose to create a 2D output image that is treated as the predicted denoising of the input view. The system must learn parameters of both the *Encoder* and *NeRF* so that any 2D view can be denoised by predicting a camera and rendering the scene. The NeRF rendering process is differentiable with respect to rays shot from the camera, which themselves depend on the camera-to-world transformation matrix produced by the encoder. All modules are end-to-end trainable, and the system is optimized by the simple MSE loss on denoising.

manipulation of pre-trained models (*e.g.,* extracting features, fine-tuning, or prompting) the only strategy for applying generative learning to solve downstream tasks.

Yuan and Maire [2023] demonstrate an alternative strategy that utilizes a generative model and relies solely on a generative learning objective, yet directly solves a downstream task (image segmentation) as a byproduct of training the generative model. Their strategy is to constrain the architecture of the generative model such that it must synthesize an image by first predicting a segmentation and then generating the corresponding image regions in parallel. Trained as a Denoising Diffusion Probabilistic Model (DDPM) [Ho et al., 2020], segmentation emerges as the bottleneck representation in a network that first partitions a noisy input into regions and then denoises each region in parallel.

We port this general concept to the problem of multiview 3D reconstruction from unknown pose, where we devise an internal pose prediction network and a NeRF comprising the task-specific architecture encapsulated within our DDPM; see Figure 3.1. We solve a small-scale generative modeling problem: learning to generate images in the collection of 2D views of a single scene.

Training examples are noised 2D images (views), the DDPM output is a predicted denoised image, and the loss is the denoising objective. Inside our generative wrapper, the model architecture dictates that we map a noisy image to a predicted camera pose and then render the NeRF from that pose to synthesize the clean output image. Successfully performing denoising in this manner requires that: (a) the NeRF stores a 3D scene representation consistent with all of the 2D views, and (b) the pose prediction network implicitly solves the 2D view correspondence problem by mapping each 2D input image to camera coordinates from which it is reconstructed by rendering the NeRF.

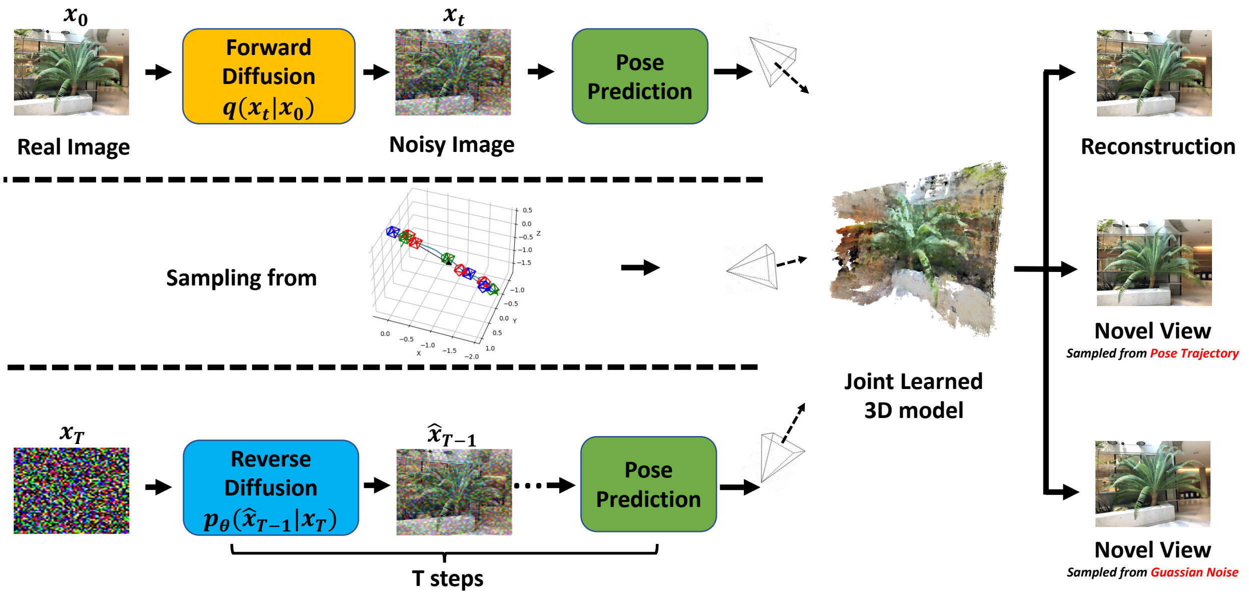


Figure 3.2: **Unifying pose prediction, 3D reconstruction, and novel-view image generation.** Our trained system (Figure 3.1) can be deployed for multiple tasks. ***Pose prediction (top):*** We can predict the pose of a previously unseen real image by adding a small amount of noise (forward diffusion) and feeding it to our *Encoder* (Fig 3.1, *left*). Rendering our learned *NeRF* from that camera pose should reconstruct the real image. ***Direct NeRF usage (middle):*** Our learned *NeRF* can be extracted and directly used to render the scene (*e.g.,* along a manually specified camera path). ***Sampling cameras and views (bottom):*** Performing sequential diffusion denoising from pure Gaussian noise input synthesizes a camera pose from which rendering the *NeRF* generates a novel view of the scene.

Figure 3.2 illustrates how our trained system jointly solves pose prediction and 3D reconstruction. Our system enables predicting the 3D pose for new (unseen) images, and re-rendering the learned scene from different camera poses which can be generated from noise or provided explicitly. As

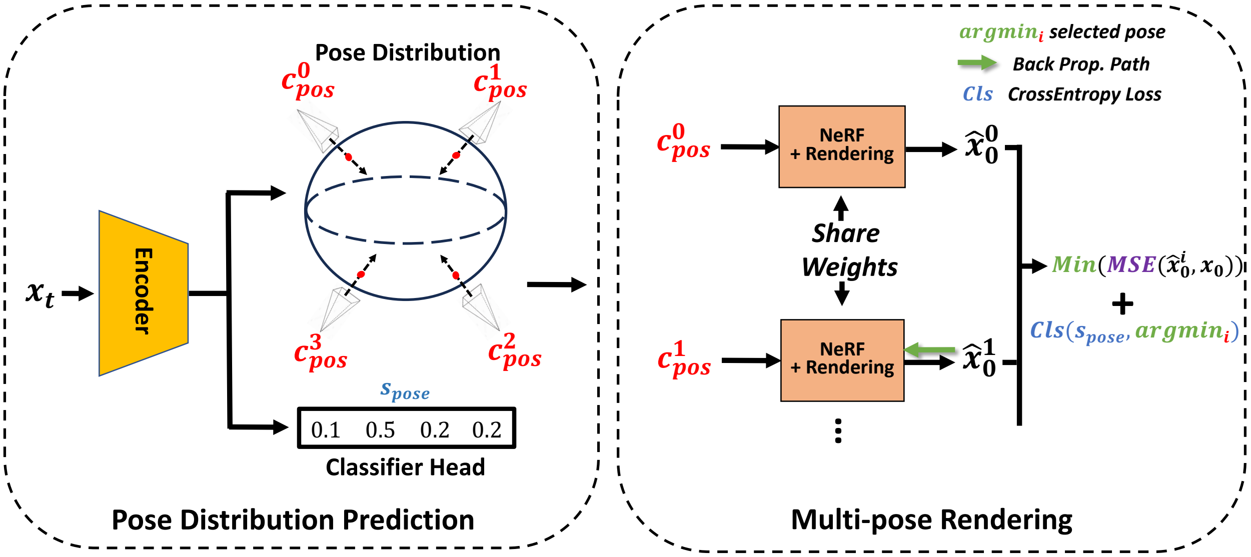


Figure 3.3: **Pose distribution representation and multi-pose rendering for 360° scenes.** In order to perform view denoising by learning a NeRF and predicting the pose from which to render it, our system (Figure 3.1) must implicitly solve multiview correspondence by mapping training images (of unknown pose) into consistent locations in the 3D environment. We enable training via gradient descent to discover such solutions for challenging multiview datasets (*e.g.,* spanning 360°) by augmenting our architecture with the capacity to represent uncertainty over a pose distribution. ***Left:*** Our encoder, given a noisy image $\boldsymbol{x}_t$, predicts parameters for multiple cameras and a corresponding probability distribution over cameras, s_{pose}. ***Right:*** During training, we render the NeRF from each predicted camera and use the best reconstruction to calculate the denoising loss; an auxiliary classification loss pushes the predicted camera distribution to upweight the selected output. At test time, we render using only the single camera predicted as most likely by the classifier.

Figure 3.3 shows and Section 3.3 describes in detail, we significantly expand the complexity of multiview reconstruction problems our system can solve by replacing our simple pose prediction network with a more expressive version. This alternative maintains a representation of uncertainty over a distribution of multiple possible poses, which gives our system, trained from scratch by gradient descent, the implicit capacity to explore more view correspondence configurations.

Our contributions are:

- A new approach to 3D reconstruction from unknown pose based entirely on generative training. Denoising is a generative wrapper that “lifts” an architecture consisting of a forward model for pose prediction and differentiable rendering to learn view correspondence and 3D reconstruction.

Compared to an autoencoder, this fully generative wrapper benefits learned reconstruction quality.

- A novel architecture for pose prediction that enables representing uncertainty during training, allowing us to learn 3D reconstructions from 2D views of arbitrary and unknown pose.
- New capabilities for 3D NeRF reconstruction which are demonstrated through experiments on arbitrary 360-degree poses. While Wang et al. [2021] can reconstruct under certain assumptions about an unknown camera (*e.g.,* forward-facing views of the scene), they fail on image collections from unconstrained pose (*e.g.,* 360° views). Our method successfully reconstructs a NeRF and infers camera pose for these challenging datasets.

3.2 Related Work

Neural Radiance Fields (NeRFs) [Mildenhall et al., 2020] have emerged as a powerful framework for 3D scene reconstruction and view synthesis, with multiple extensions and improvements [Barron et al., 2021, Martin-Brualla et al., 2021, Barron et al., 2023]. NeRF++ [Zhang et al., 2020a] adds spatially-varying reflectance and auxiliary tasks for better training. PixelNeRF [Yu et al., 2021] extends NeRF to generate high-quality novel views from one or few input images, but still requires camera pose information. NeRF−− [Wang et al., 2021] jointly optimizes camera intrinsics and extrinsics as learnable parameters while training NeRFs. However, their proposed training scheme and camera parameterization cannot handle large camera rotation and is restricted to forward-facing views of the scene.

Generative models for 3D reconstruction aim to infer the underlying 3D structure of a scene from a set of 2D images. These models often learn a latent representation of the 3D scene and use it to generate novel views or perform other tasks such as object manipulation or scene editing. Generative Radiance Fields (GRAFs) [Schwarz et al., 2020] combine NeRFs with VAEs or GANs to generate novel 3D scenes without explicit 3D geometry. GRAFs learn a latent space encoding scene structure, with NeRF mapping points in this space to 3D radiance fields. DiffRF [Müller et al., 2023] leverages the diffusion prior to perform 3D completion in a two-stage manner, which is further improved by SSD-NeRF [Chen et al., 2023a] with a single stage training scheme and an end-to-end objective that jointly optimizes a NeRF and diffusion. Multiple works combine NeRF with generative models for the purpose of 3D synthesis [Chan et al., 2021, Meng et al., 2021, Gu et al., 2021], including ones that place NeRF and diffusion models in series [Poole et al., 2022, Lin et al., 2023, Wang et al., 2023a]. Our framework's nested structure differs, as our aim is not to learn to generate novel 3D scenes; we aim to use generative training to solve the classic multiview 3D reconstruction problem.

Pose estimation is the challenging task of estimating object or camera position and orientation within a scene. COLMAP [Schönberger and Frahm, 2016, Schönberger et al., 2016] uses a

Structure-from-Motion (SfM) [Schönberger and Frahm, 2016] approach for pose estimation, can handle challenging scenes with varying lighting conditions and viewpoints, and is widely used in NeRF training. However, this collection of techniques requires a large number of images for accurate pose estimation; pre-processing also restricts flexibility. PoseDiffusion [Wang et al., 2023b] and Camera-as-Rays [Zhang et al., 2024] use a diffusion model to denoise camera parameters and rays. Although sharing similar spirit in adopting diffusion, these methods require a supervised pertaining stage. More importantly, diffusion serves as a different role in our model: instead of denoising cameras to recover the pose distribution, we modulate a pose prediction system embedded inside the diffusion training process, yielding pose information as a latent representation.

3.3 Method

3.3.1 Unsupervised Pose Prediction from a Single Image

Our pose module (Figure 3.1, left) consists of several components designed to predict, from a 2D image, the position and orientation of a camera in the scene. We design the encoder based on a standard DDPM U-Net. We obtain input $\boldsymbol{x}_t$, a noise version of $\boldsymbol{x}_0$, via forward diffusion:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) := \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)I),$$
$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(0,1), \tag{3.1}$$

where $\alpha_t = 1 - \beta_t, \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_t$. We encode the pose information in the form of a camera-to-world transformation matrix, $T_{wc} = [Ro|ts]$, where $Ro \in SO(3)$ represents the camera's rotation, and $ts \in \mathbb{R}^3$ represents its translation. Following [Wang et al., 2021], we adopt Rodrigues' formula to form the rotation matrix Ro from axis-angle representation:

$$Ro = I + \sin(\phi)[\omega]_\times + (1-\cos(\phi))[\omega]_\times^2 \tag{3.2}$$

where ϕ is the rotation angle, ω is a normalized rotation axis, and $[\omega]_\times$ is the skew-symmetric matrix of the rotation axis vector ω.

Different from [Wang et al., 2021], which formulates ω_i and translation ts_i as trainable parameters, for each input image $\boldsymbol{x}^i$, our system maps the corresponding U-Net encoder features to two 3-dimensional input-dependent feature vectors. By doing so, we not only learn to fit the pose information during training, but also obtain a pose predictor network applicable to any input 2D image.

3.3.2 3D Optimization with Denoising Rendering

With the predicted camera pose from the U-Net Encoder, the system is able to perform denoising via differentiable rendering. Specifically, as shown in the right side of Figure 3.1, we sample the differentiable coordinates $\boldsymbol{c}_{pos}$, which are fed into a NeRF MLP model to learn object density and opacity, generating a 2D image reconstruction $\hat{\boldsymbol{x}}_0$.

Benefiting from the compatibility between NeRF reconstruction loss and DDPM denoising objective, we train our model end-to-end by simply minimizing the pixel-wise distance from $\hat{\boldsymbol{x}}_0$ to $\boldsymbol{x}_0$. Model weights of the camera predictor (U-Net encoder) and NeRF MLPs are optimized with loss:

$$L = \mathbb{E}||\hat{\boldsymbol{x}}_0 - \boldsymbol{x}_0||_2^2 \tag{3.3}$$

Algorithm 3 summarizes training.

3.3.3 Multi-pose Rendering for Scene Reconstruction from 360° Views

A failure to estimate poses accurately can occur when rotation perturbations exceed a certain threshold in Eq. 3.2, which prevents learning from 360° views of a scene [Wang et al., 2021]. Even with good reconstruction in 2D space, an overfitting issue can occur during optimization, where NeRF compensates by creating multiple disjoint copies of scene fragments instead of a unified 3D reconstruction. We address these issues via a higher capacity pose predictor capable of representing uncertainty (Figure 3.3).

Pose distribution prediction. A simple camera parameterization is to restrict position to a fixed-radius sphere with fixed intrinsics, and the constraint of always looking towards the origin at (0, 0, 0). Assuming the object in a 360° scene is centered, rotated, and scaled by some canonical alignment, such a parametrization has only two degrees of freedom. However, this simplistic approach restricts model capacity for capturing diverse viewpoints or extensive rotations.

Algorithm 3 Generative Lifting to 3D:
Single Camera Pose Prediction & NeRF

```
Input: Multiview image collection $\boldsymbol{\mathcal{X}}$
Output: Encoder (pose predictor) & NeRF
Initialize: Model weights $\theta$, Timesteps $T$
for iter $= 1$ to Iter$_{total}$ do
    Sample $\boldsymbol{x}_0 \in \boldsymbol{\mathcal{X}}$, $t \in [1, T]$.
    Sample $\boldsymbol{x}_t$ using Eq. 3.1.
    Predict $Ro$ using Eq. 3.2 and $ts$.
    Compute $\hat{\boldsymbol{x}}_0$ by rendering the NeRF from the pose predicted for $\boldsymbol{x}_t$ (see Figure 3.1).
    Backprop from loss in Eq. 3.3.
    Update model weights.
end for
return $\theta$
```

Algorithm 4 Generative Lifting to 3D using Pose Distribution Modeling & Multi-pose NeRF Rendering

```
Input: Multiview image collection $\boldsymbol{\mathcal{X}}$
Output: Encoder (multi-pose predictor & classifier) & NeRF
Initialize: Model weights $\theta$, Timesteps $T$, and initial pose of candidate cameras
for iter $= 1$ to Iter$_{total}$ do
    Sample $\boldsymbol{x}_0 \in \boldsymbol{\mathcal{X}}$, $t \in [1, T]$.
    Sample $\boldsymbol{x}_t$ using Eq. 3.1.
    Predict poses and corresponding $\boldsymbol{s}_{pose}$, as in Figure 3.3.
    Compute $\{\hat{\boldsymbol{x}}_0^i\}$ using multi-pose rendering (Figure 3.3).
    Backprop along the path of the best render via Eq. 3.4.
    Update model weights.
end for
return $\theta$
```

We propose a more flexible approach that allows for a wider range of camera positions and orientations. Given a 2D image captured from a specific viewpoint, instead of predicting a single transformation, we sample the camera's position from a distribution of multiple cameras that cover a larger range of positions and orientations on a sphere.

As Figure 3.3 shows, different candidate cameras spread over the sphere, pointing to the origin at initialization. Each input view predicts parameters for all cameras in the distribution. An auxiliary classifier head predicts the probability of the input view corresponding to each camera in the distribution. Early in training, such a design facilitates searching over multiple pose hypotheses in order to discover a registration of all views into a consistent coordinate frame. Only one camera prediction per input view need be correct, as long as the system also learns which one.

Joint optimization with multi-pose rendering. We render the NeRF separately from each candidate camera to produce a set of 2D images $\{\hat{\boldsymbol{x}}_0^i\}$. During backpropagation, we only allow the gradient to pass selectively to optimize the best match between the true image and the rendered reconstruction. The selected branch index serves as a pseudo-label to co-adapt the classifier head in a self-supervised bootstrapping manner. The total loss for joint training is:

$$L = \min_i ||\hat{\boldsymbol{x}}_0^i - \boldsymbol{x}_0||_2^2 + \lambda CrossEntropy(\boldsymbol{s}_{pose}, \arg\min_i ||\hat{\boldsymbol{x}}_0^i - \boldsymbol{x}_0||_2^2) \tag{3.4}$$

where λ is the trade-off parameter between view reconstruction and camera classification. We set λ as 0.1 in experiments and investigate its effect in an ablation study. Algorithm 4 summarizes training.

3.3.4 Novel View Generation

Figure 3.2 illustrates the different modalities in which our trained system can be used.

Pose prediction & reconstruction. Given input image $\boldsymbol{x}_0$, we sample a noisy version $\boldsymbol{x}_t$ through forward diffusion in Eq. 3.1. We then pass $\boldsymbol{x}_t$ to the model. Our system estimates the camera pose of $\boldsymbol{x}_t$ with respect to the scene and recovers a clean image reconstruction with one-step denoising.

Sampling from pose trajectory. Though not trained with any camera pose information, our system can generate novel views using a pre-defined camera trajectory, acting as a conventional NeRF model.

Sampling from Gaussian noise. A unique property of our system is its support for sampling cameras and scene views. Using reverse diffusion, our model can generate a realistic novel view and the corresponding camera pose, starting from a pure noise input $\boldsymbol{x}_T \sim \mathcal{N}(0, 1)$. We perform T steps of reverse diffusion (predict $\boldsymbol{x}_{t-1}$ from $\boldsymbol{x}_t$) to progressively generate a novel view.

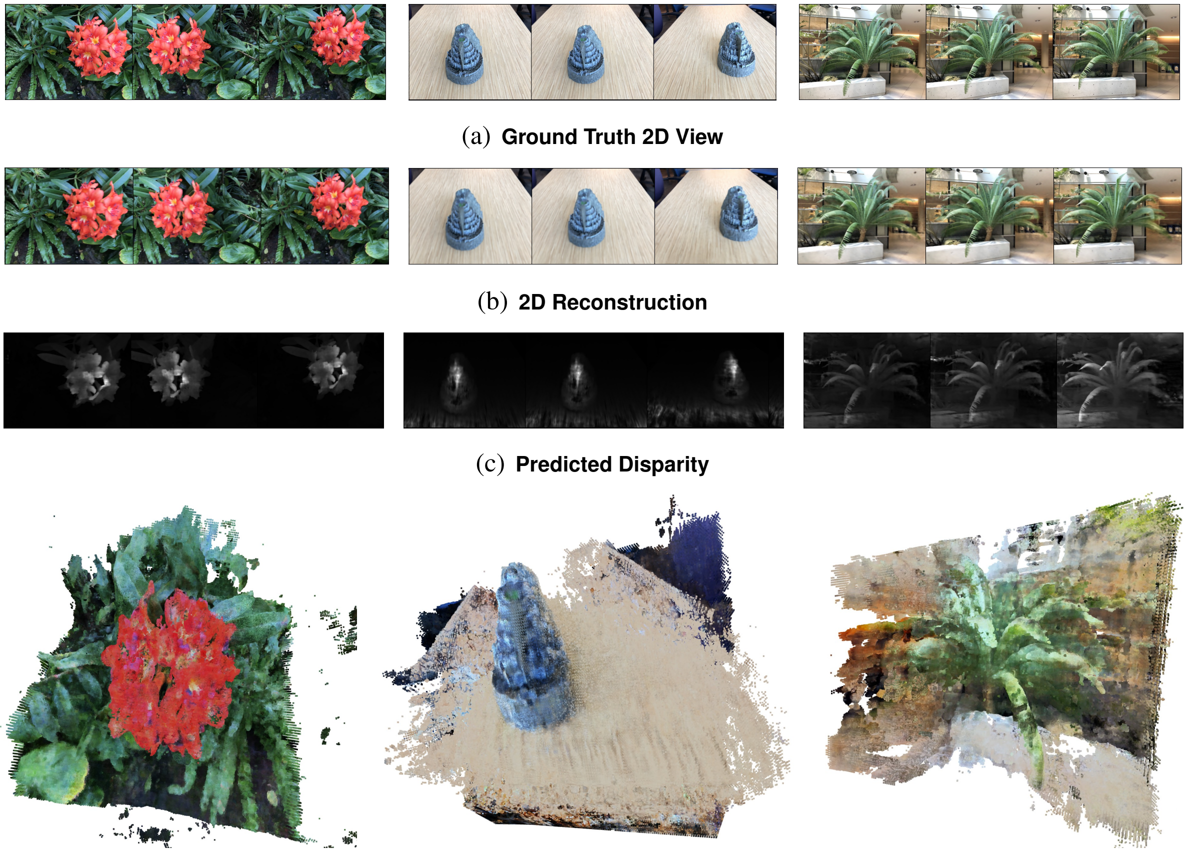
3.4 Experiments

We conduct the experiments on the face-forwarding dataset LLFF Mildenhall et al. [2019] with a resolution of 378×504, using the single camera prediction system depicted in Alg. 3. To handle 360° scenes, we adopt the multi-pose rendering as in Alg. 4 on ShapeNet Car Chang et al. [2015] (5 scenes), Lego and Drums Mildenhall et al. [2020] with a resolution of 128×128. Instead of simultaneously optimizing two networks: one 'coarse' and one 'fine', we only use a single network to represent 360° scenes for all methods. We initialize 8 camera candidates spread over 8 quadrants of a sphere for the ShapeNet Car scene, and 12 candidates over 4 quadrants of a semi-sphere for Lego and Drums.

3.4.1 Implementation Details

For all experiments, we use a U-Net Ronneberger et al. [2015] encoder as the pose prediction module. The downsampling stack performs five steps of downsampling, each with 2 residual blocks. From highest to lowest resolution, U-Net stages use $[C, C, 2C, 2C, 4C]$ channels, respectively. We set $C = 64$ for all models. Figure 3.12 details the network architecture. We use 100 denoising steps for all models.

Our method involves two sets of trainable parameters: NeRF model weights and pose prediction network weights; we adopt separate Adam optimizers, with learning rates $1e^{-4}$ and $2e^{-5}$ for NeRF and pose prediction, respectively. We set (β_1, β_2) as $(0.9, 0.999)$ for both optimizers. We adopt the same batch size and learning rate scheduler used to train the corresponding baseline NeRF as in Mildenhall et al. [2020]. We train all models for 200k iterations. Code segments 3.1, 3.2 and 3.3 detail our camera parameterization.



(a) **Ground Truth 2D View**

(b) **2D Reconstruction**

(c) **Predicted Disparity**

(d) **Point Cloud from NeRF**

Figure 3.4: **Reconstructions of images unseen during training on three scenes from LLFF [Mildenhall et al., 2019].**

3.4.2 Multi-view 3D Reconstruction

We measure the quality of reconstructions obtained by the *top* pipeline in Figure 3.2. We directly input previously unseen images from different views to generate reconstructions.

LLFF dataset. Figure 3.4 shows we obtain good-quality reconstructions and disparity predictions. Point cloud visualization plots the density and opacity output from our NeRF model at 3D coordinates.

360° **scenes.** Camera motions with large rotation perturbations cause failures in NeRF--; it cannot handle 360° scenes like Lego. Our method solves this challenging case from a single input image, without relative pose estimation between image pairs. To obtain reconstructions, we

Table 3.1: **Multiview reconstruction quality (PSNR, SSIM, & LPIPS).** Our system, without pose knowledge, reconstructs 3D scenes from challenging image collections (views spanning 360°) on which NeRF-- [Wang et al., 2021] fails. Supervised denotes standard NeRF training using ground-truth camera pose.

		PSNR (↑)			SSIM (↑)			LPIPS (↓)		
Type	Scene	Supervised	NeRF--	Ours	Supervised	NeRF--	Ours	Supervised	NeRF--	Ours
Forward-Facing	Fern	22.22	21.67	17.02	0.64	0.61	0.42	0.47	0.50	0.55
	Flower	25.25	25.34	22.42	0.71	0.71	0.58	0.36	0.37	0.43
	Fortress	27.60	26.20	22.02	0.73	0.63	0.50	0.38	0.49	0.51
	Horns	24.25	22.53	17.48	0.68	0.61	0.43	0.44	0.50	0.55
	Leaves	18.81	18.88	14.44	0.52	0.53	0.42	0.47	0.47	0.60
	Orchids	19.09	16.73	14.34	0.51	0.39	0.40	0.46	0.55	0.58
	Room	27.77	25.84	22.36	0.87	0.84	0.48	0.40	0.44	0.49
	Trex	23.19	22.67	19.96	0.74	0.72	0.62	0.41	0.44	0.51
360°	Car	28.98	✗	26.43	0.95	✗	0.92	0.08	✗	0.08
	Lego	25.44	✗	21.38	0.92	✗	0.86	0.09	✗	0.12
	Drums	22.12	✗	18.65	0.89	✗	0.82	0.08	✗	0.16

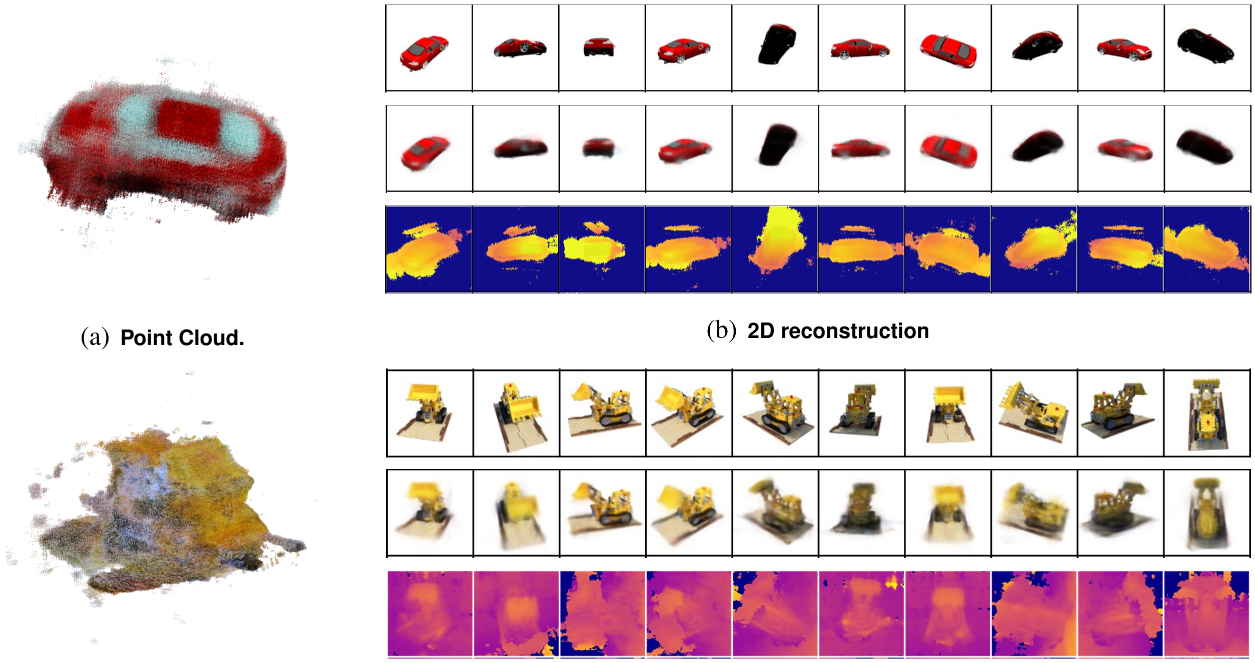
determine the camera pose for the input image based on the maximum score of the classification head in Figure 3.3 before rendering. As Figure 3.5 shows, we generate good reconstructions and point clouds.

To quantify the quality of our reconstructions, we compare the PSNR, SSIM, LPIPS Zhang et al. [2018] with supervised NeRF (using pre-processed pose information), and NeRF--. As Table 3.1 shows, for face-forwarding scenes, our method achieves reasonably high performance. We cannot beat NeRF-- because we aim to solve a more general pose prediction problem from a single input, instead of fitting camera parameters as trainable variables. For 360° scene views, which NeRF-- completely fails to handle, our method still yields good reconstructions. This validates the design choice of our multi-pose rendering system in tolerating large camera pose perturbations.

3.4.3 Visualization of Pose Optimization

As shown in Figure 3.6, we also demonstrate that our pose prediction system can generate reasonable pose estimates, though not in the same coordinate system, compared with ground-truth cameras.

Camera distribution evolution during optimization To better demonstrate the pose prediction



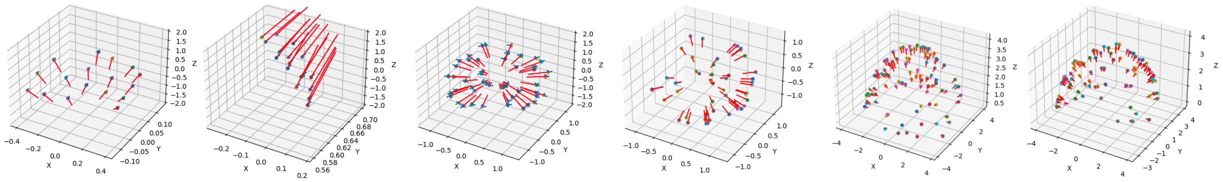
(a) **Point Cloud.**

(b) **2D reconstruction**

(c) **Point Cloud.**

(d) **2D reconstruction**

Figure 3.5: **Reconstructions on** 360° **scenes.**



(a) **Fern GT.**

(b) **Fern Pred.**

(c) **Car GT.**

(d) **Car Pred.**

(e) **Lego GT.**

(f) **Lego Pred.**

Figure 3.6: **Visualization of camera poses for Fern, Car, and Lego.**

refinement during the optimization process, we visualize the camera poses at different training iterations for the Car scene. As shown in Figure 3.7, the candidate poses refers to all possible predictions over 8 quaternions while the selected poses represent those with maximum classification scores. During training, the learned candidate poses tend to cover the sphere uniformly, with the selected pose distributions gradually converge to that provided by the pre-processed dataset. We observe simultaneous refinements of both 3D model and pose prediction along the training process.

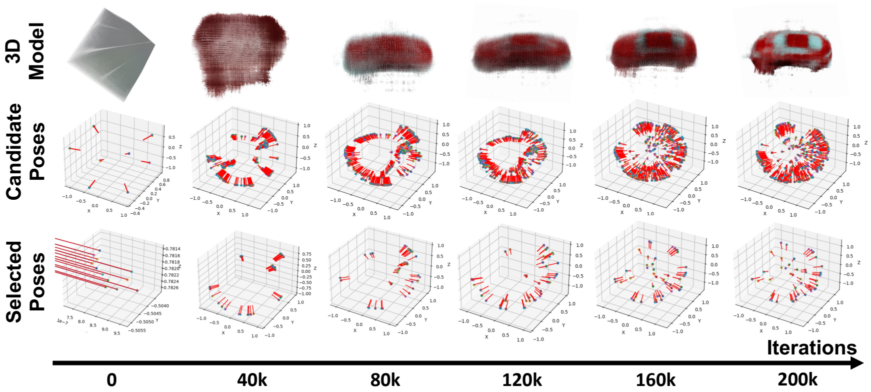


Figure 3.7: **Joint 3D and pose optimization for Car during training.**

λ	Acc.	PSNR
0.01	28	✗
0.02	72	19.43
0.05	66	18.88
0.1	98	26.43
0.2	90	22.19
0.5	100	✗
1.0	100	✗

Table 3.2: **Effect of λ.**

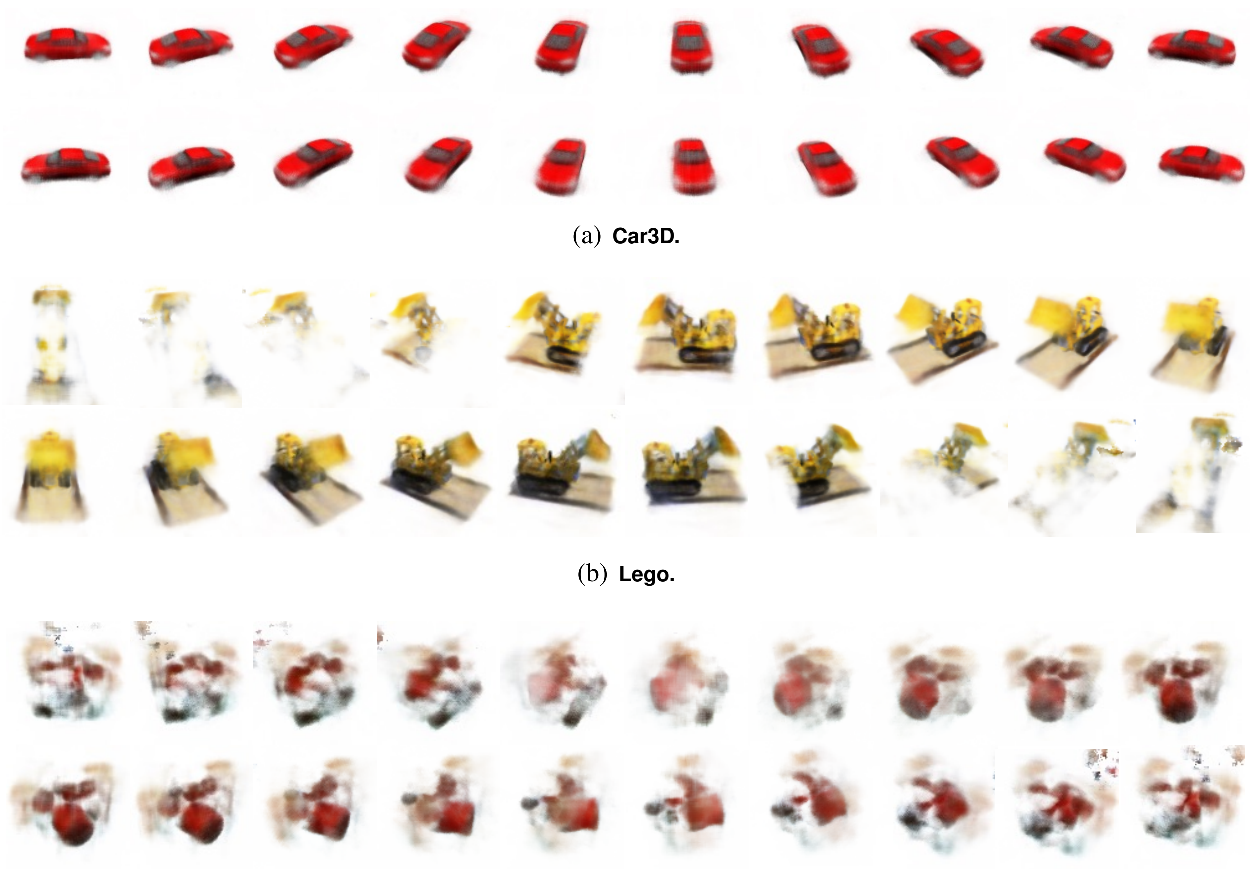
3.4.4 Novel View Synthesis

From camera trajectory: We show that our joint-learned 3D model, even learned with unknown pose, can generate novel views using a manually designed camera trajectory as in a supervised NeRF. In Figure 3.8, we generate novel views using a continuous spiral camera path (pointing to the origin) for three different challenging scenes.

From Gaussian noise: Given the nice property of denoising diffusion training, we have the flexibility to generate novel views from Gaussian noise progressively. In the DDPM Markov process, the model implicitly formulates a mapping between noise and data distributions. We validate this by visualizing the reconstructed $\hat{x}_{t-1}$ and $\hat{x}_0$ along the sequential reversed diffusion process in Figure 3.9. We observe gradual refinement as denoising steps approach $t = 0$.

3.4.5 Abalation Studies

Training with clean images. Training NeRF with denoising diffusion is an essential part of our method. We show the effectiveness of this design by varying the input with clean images in our method, in which case our architecture downgrades to an autoencoder (AE). As shown in Figure 3.10, the baseline trained with clean images (denoted as AE) yields an incorrect 3D model for the Car scene. This suggests the failure of camera pose prediction, hence leading to a NeRF over-fitting issue. Moreover, AE fails to perform novel-view synthesis given a pre-defined camera



(a) Car3D.

(b) Lego.

(c) Drums.

Figure 3.8: Novel view synthesis from circle trajectory.

trajectory, as shown in Figure 3.13. Thus, denoising diffusion training not only provides a new way to perform novel view synthesis, but also significantly improves the learned 3D reconstruction.

Candidate camera numbers in multi-pose rendering. For Lego and Drums on a semi-sphere, we choose to use 12 candidate cameras instead of 4 at initialization for each view. The 4-way case poses an easier camera classification task to the system, but restricts flexibility for discovering view correspondence and thereby pushes the NeRF to overfit on incorrect pose predictions. Increasing capacity to 12 cameras during training addresses this issue and prevents the system from converging to a suboptimal solution. As shown in Figure 3.11, $4\times$ and $8\times$ variants fail to converge to the correct pose distributions, while the $12\times$ succeeds.

Trade-off between classification and reconstruction. Due to the discrepancy between training and

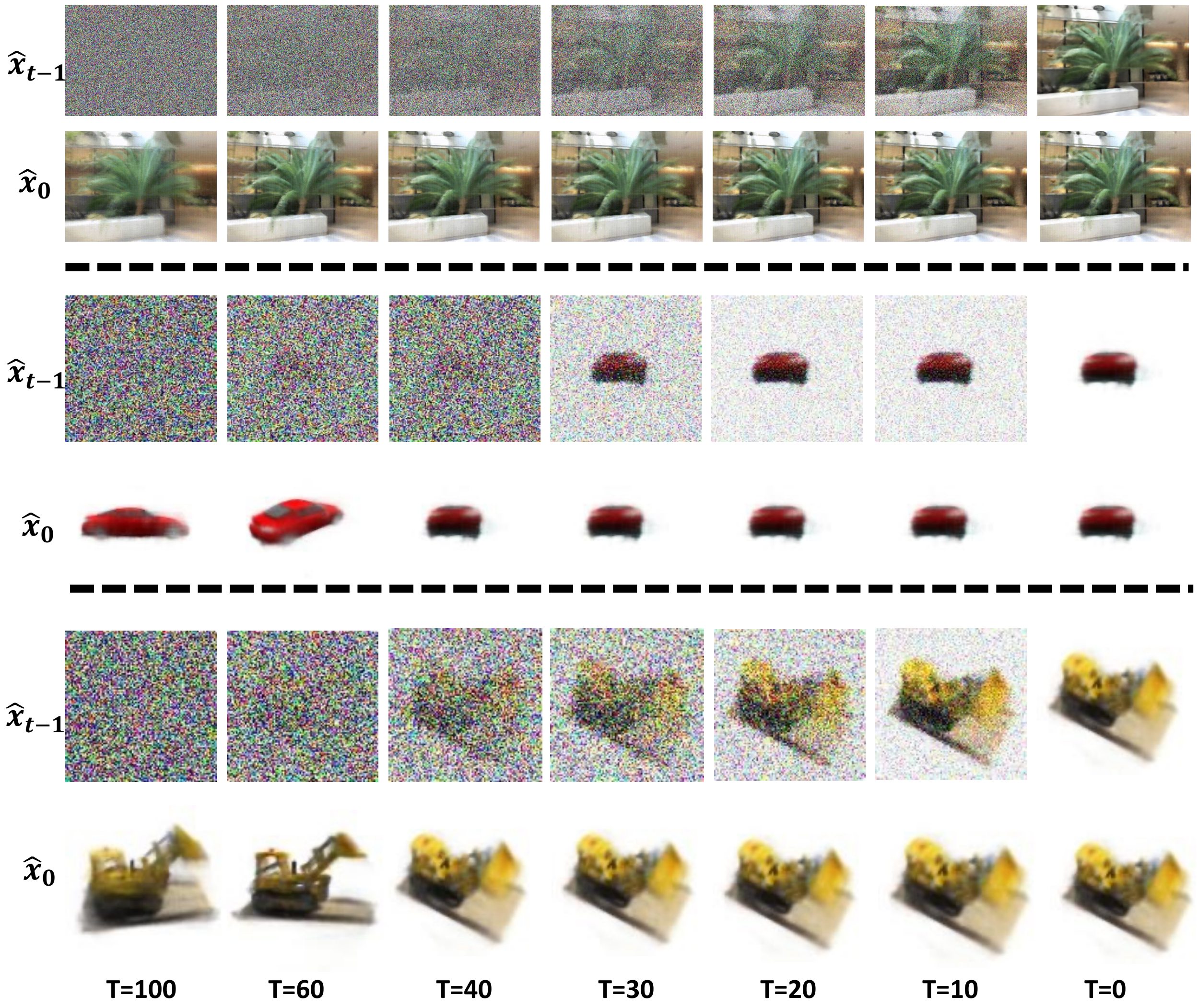
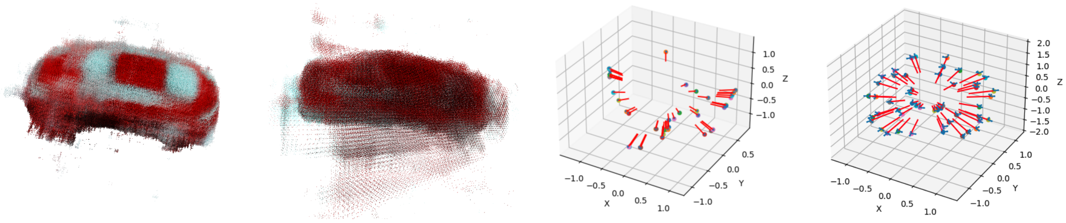


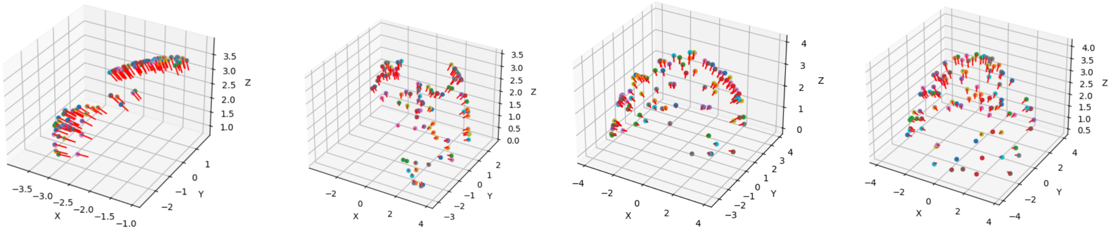
Figure 3.9: **Novel view synthesis from Gaussian noise.**

testing in our multi-pose system, the quality of rendering highly relies on the camera classification accuracy. To evaluate the performance of this self-supervised classifier, we use the camera index which produces the minimum reconstruction loss as ground-truth. We study the effect of the classification loss term by alternating $\lambda \in \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0\}$, generating trade-offs between the accuracy and PSNR for Car. As shown in Table 3.2, NeRF training yields a poor reconstruction and even an optimization failure when the classification accuracy is low. When we set λ as a large value, the model runs into a local minimum: the classifier obtains 100% accuracy at early training iterations and poses cannot be jointly optimized to discover correspondence.



(a) **Ours** (b) **AE** (c) **AE-Pred** (d) **GT**

Figure 3.10: **Learning w/o denoising diffusion.**



(a) **4**× (b) **8**× (c) **12**× (d) **GT**

Figure 3.11: **Poses learned with different camera number.**

3.5 Appendix

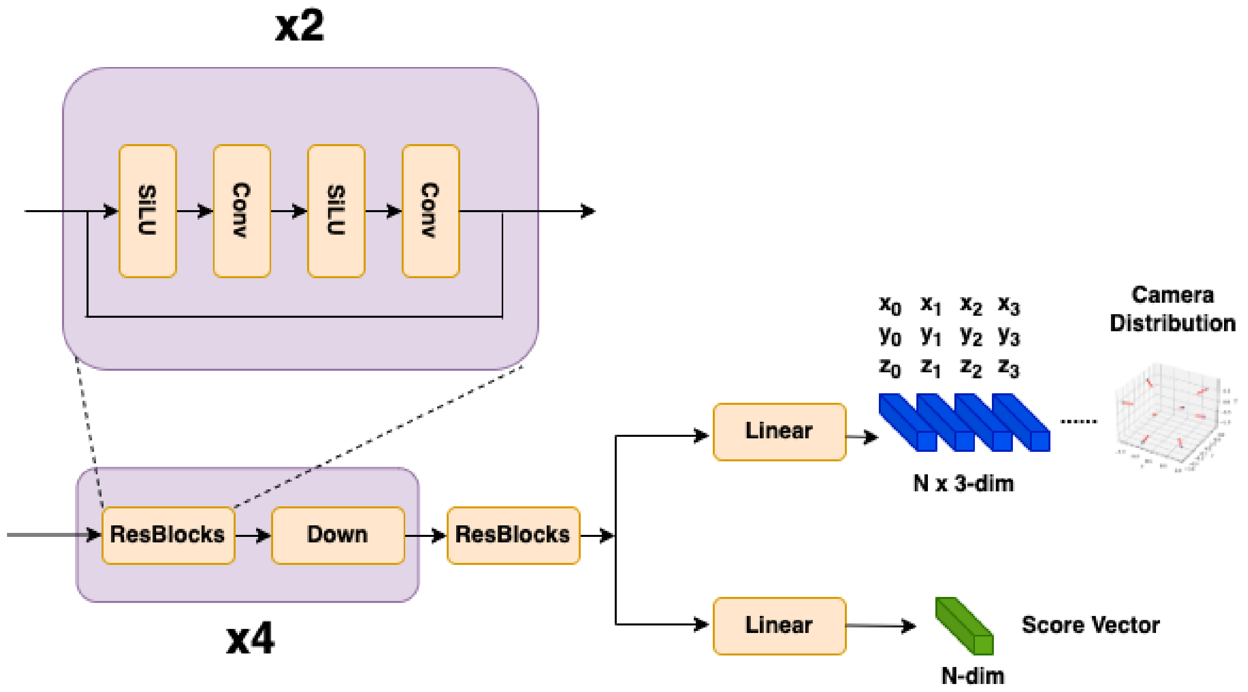


Figure 3.12: **Detailed architecture of pose prediction network.** A shared encoder trunk processes an input image and branches into heads for predicting a set of candidate camera poses, as well as a score vector indicating the probability the image was acquired from each of the predicted cameras.

```
import torch as th
import pytorch3d.transforms as tf
def gen_rotation_matrix_from_xyz(xyz, in_plane=th.from_numpy(np.array([0.0, 0.0, 0.0])).cuda().float()):
    cam_from = xyz
    cam_to = th.from_numpy(np.zeros(3)).float().cuda()
    tmp = th.from_numpy(np.array([0.0, 1.0, 0.0])).float().cuda()

    diff = cam_from - cam_to
    forward = diff / th.linalg.norm(diff)
    crossed = th.cross(tmp, forward)
    right = crossed / th.linalg.norm(crossed)
```

```python
    up = th.cross(forward, right)

    R = th.stack([right, up, forward])
    R_in_plane = tf.rotation_conversions.euler_angles_to_matrix(in_plane, "XYZ")
    return R_in_plane @ R
```

Code 3.1: Obtain rotation matrix from camera position

```python
import torch as th
def lkat(eye, target, up):
    forward = normalize(target - eye)
    side = normalize(th.cross(forward, up))
    up = normalize(th.cross(side, forward))

    zero = th.zeros(1).float().cuda()
    one = th.ones(1).float().cuda()
    trans_v0 = th.cat([side[0:1], up[0:1], -forward[0:1], zero])  # (3, 1)
    trans_v1 = th.cat([ side[1:2], up[1:2], -forward[1:2],zero])
    trans_v2 = th.cat([side[2:3], up[2:3], -forward[2:3], zero])
    trans_v3 = th.cat([ -th.dot(side, eye)[None], -th.dot(up, eye)[None], th.dot(forward, eye)[None], one])
    c2w = th.stack([trans_v0, trans_v1, trans_v2,trans_v3], dim=0)
    return c2w
```

Code 3.2: Camera transformation with pointing to the origin.

```python
import torch as th
eye_candidates = th.Tensor([[1,1,1],[1,-1,1],[-1,1,1],[-1,-1,1],
                            [1,1,1],[1,-1,1],[-1,1,1],[-1,-1,1],
                            [1,1,1],[1,-1,1],[-1,1,1],[-1,-1,1]]).cuda()
r = th.tensor([4.0]).float().cuda()
target = th.from_numpy(np.zeros(3)).float().cuda()
up = th.from_numpy(np.array([0.0, 1.0, 0.0])).float().cuda()
```

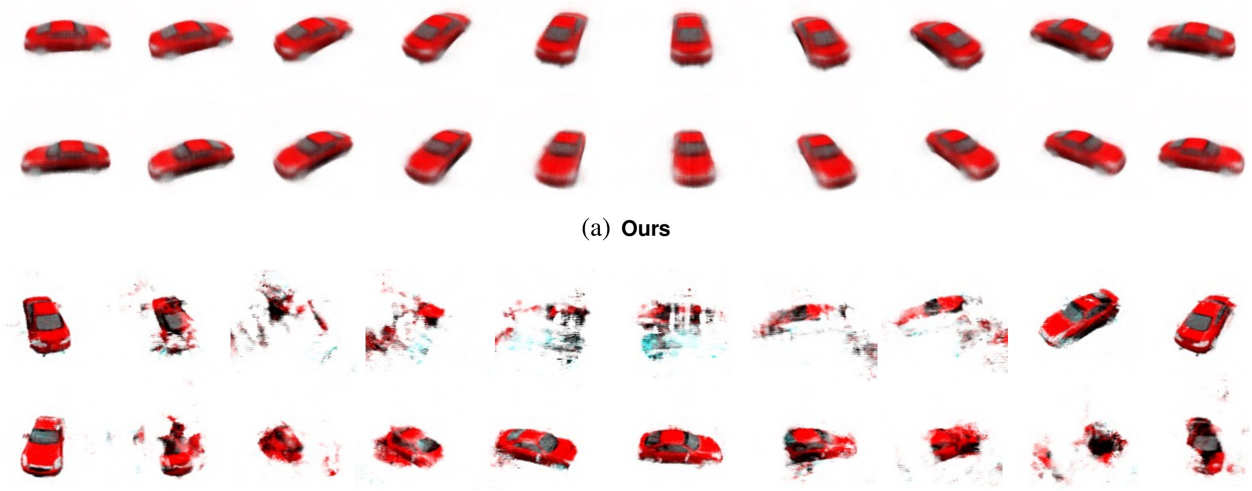
```python

# h: the output of the last residual block in pose prediction model.
h1 = linear1(h.squeeze(-1)).squeeze() # (12*3, )
h2 = linear2(h.squeeze(-1)).squeeze() # (12, ), score vector

zero = th.zeros(1).float().cuda()
init_cam_pos = th.cat([ zero,    zero,   r)  # (3, 1)
all_poses = []

for index in range(12):
    h3= th.sigmoid(h1[3*index:index*3+3])
    h3 = th.diag(eye_candidates[index])@h3
    h3 = h3/th.linalg.norm(h3)
    R1 = gen_rotation_matrix_from_xyz((h3)
    eye = (R1 @ init_cam_pos)
    look_at1 = lkat(eye, target, up)
    pose1 = th.eye(4).float().cuda()
    pose1[:3, :3] = look_at1[:3, :3]
    pose1[:3, 3] = -look_at1[:3, :3] @ look_at1[3, :3]
    all_poses.append(pose1[:3,:4])
all_poses = th.stack(all_poses, 0)
return all_poses, h2 #pose distribution (12,3,4), scores (12, 1)
```

Code 3.3: Pose distribution prediction with 12× camera candidates



(a) **Ours**

(b) **AE Baseline**

Figure 3.13: **Diffusion training benefits novel view synthesis on Car3D.** Our system, wrapped within a DDPM for training, significantly outperforms the same architecture trained as a simple autoencoder (AE). Training with the more challenging denoising task yields more robust generalization for the pose prediction network and NeRF scene representation.

3.6 Summary

We propose a novel technique that places NeRF inside a probabilistic diffusion framework to accurately predict camera poses and create detailed 3D scene reconstructions from collections of 2D images. Our approach enables training NeRF from images with unknown pose. Using a carefully constrained architecture and differentiable volume renderer, we learn a camera pose predictor and 3D representation jointly. Our experimental results and ablation studies confirm the effectiveness of this method, demonstrating its capability to produce high-quality reconstructions, localize previously unseen images, and sample novel-view images, all while trained in an entirely unsupervised manner.

CHAPTER 4

CONCLUSION

This thesis presents a new approach to generative learning by proposing to design domain-specific architectures that directly solve end tasks when trained with the sole objective of denoising, without reliance on annotated data. These architectures utilize a computational bottleneck that specifically caters to the needs of the task and the data involved, allowing for learning processes that directly address the challenges posed by unsupervised learning scenarios. This stands in contrast to traditional approaches, which often depend on large-scale pre-trained models fine-tuned on specific datasets.

The effectiveness of this approach is demonstrated through rigorous experimentation across various tasks, showcasing the model's versatility and robustness. Notably, the thesis reports success in unsupervised image segmentation and generation of synthetic images. This is a significant achievement as it proves the model's capacity to perform complex tasks without labeled data.

Furthermore, we apply this architectural design strategy to produce an innovative solution to the challenging problem of pose estimation and 3D reconstruction from 2D images. Our successful unsupervised learning of camera poses and 3D geometry, within a diffusion framework, highlights the potential of this approach to impact areas such as virtual reality, augmented reality, and robotic navigation, where understanding 3D spaces from 2D inputs is crucial.

Overall, this thesis validates the effectiveness of combining domain and task-specific computational architectures with generative training objectives to learn interpretable, structured latent representations. This research paves the way for further exploration and development in unsupervised learning, potentially transforming how computer vision tasks are approached in the future.

REFERENCES

Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building Rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

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

Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *ICLR*, 2022.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CoRR*, abs/2111.12077, 2021.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *ICCV*, 2023.

Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *CVPR*, 2015.

Adam Bielski and Paolo Favaro. Emergence of object segmentation in perturbed generative models. In *NeurIPS*, 2019.

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

Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

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

Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, 2021.

Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. 2015.

Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *ICCV*, 2023a.

Mickaël Chen, Thierry Artières, and Ludovic Denoyer. Unsupervised object segmentation by redrawing. In *NeurIPS*, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020a.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, 2016.

Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020b.

Yida Chen, Fernanda Viégas, and Martin Wattenberg. Beyond surface statistics: Scene representations in a latent diffusion model. *NeurIPS workshop*, 2023b.

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

Lee Raymond Dice. Measures of the amount of ecologic association between species. *Ecology*, 1945.

Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

Xiaodan Du, Nicholas I. Kolkin, Greg Shakhnarovich, and Anand Bhattad. Generative models: What do they know? do they know things? let's find out! *CoRR*, abs/2311.17137, 2023.

Dave Epstein, Taesung Park, Richard Zhang, Eli Shechtman, and Alexei A. Efros. Blobgan: Spatially disentangled scene representations. In *ECCV*, 2022.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022.

Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

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

Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

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

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NeurIPS*, 2017.

R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

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

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

Jyh-Jing Hwang, Stella X. Yu, Jianbo Shi, Maxwell D. Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *ICCV*, 2019.

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

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2014.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

Iasonas Kokkinos. Pushing the boundaries of boundary detection using deep learning. In *ICLR*, 2016.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, 2017.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Ke Li and Jitendra Malik. Implicit maximum likelihood estimation. *CoRR*, abs/1809.09087, 2018.

Tianhong Li, Huiwen Chang, Shlok Kumar Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. MAGE: masked generative encoder to unify representation learning and image synthesis. *CoRR*, abs/2211.09117, 2022.

Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5741–5751, 2021.

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 *CVPR*, 2023.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014.

Runtao Liu, Zhirong Wu, Stella X. Yu, and Stephen Lin. The emergence of objectness: Learning zero-shot segmentation from videos. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *NeurIPS*, 2021.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *ECCV*, 2016.

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

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

David Martin, Charless Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *PAMI*, 2004.

Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021.

Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. In *ICCV*, pages 6351–6361, 2021.

Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulo, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4328–4338, 2023.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

Yassine Ouali, Céline Hudelot, and Myriam Tami. Autoregressive unsupervised image segmentation. In *ECCV*, 2020.

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

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

Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *CVPR*, 2022.

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

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

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

Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. “GrabCut”: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 2004.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

Pedro Savarese, Sunnie S. Y. Kim, Michael Maire, Greg Shakhnarovich, and David McAllester. Information-theoretic segmentation by inpainting error maximization. In *CVPR*, 2021.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: an open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *NeurIPS*, 2020.

Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Zhijiang Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *CVPR*, 2015.

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

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3D. In *ACM siggraph 2006 papers*, pages 835–846. 2006.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with PixelCNN decoders. In *NeurIPS*, 2016.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

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

Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9773–9783, 2023b.

Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. *ICCV*, 2023.

Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, 2015.

Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.

Xin Yuan and Michael Maire. Factorized diffusion architectures for unsupervised image generation and segmentation. *arXiv preprint arXiv:2309.15726*, 2023.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017.

Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. *ICLR*, 2024.

Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020a.

Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *ECCV*, 2016.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Shiyin Zhang, Jun Hao Liew, Yunchao Wei, Shikui Wei, and Yao Zhao. Interactive object segmentation with inside-outside guidance. In *CVPR*, 2020b.

Xiao Zhang and Michael Maire. Self-supervised visual representation learning from hierarchical grouping. In *NeurIPS*, 2020.

Xiao Zhang, David Yunis, and Michael Maire. Deciphering 'what' and 'where' visual pathways from spectral clustering of layer-distributed neural representations. *CoRR*, abs/2312.06716, 2023.

Zijian Zhang, Zhou Zhao, and Zhijie Lin. Unsupervised representation learning from pre-trained diffusion probabilistic models. In *NeurIPS*, 2022.