Wrocław University of Science and Technology
Faculty of Information and Communication Technology

Field of study: **Artificial Intelligence**
Speciality: **---**

MASTER THESIS

A Method for Image Generation Using Conditional Multi-Views

Eryk Wójcik

Supervisor
dr Kamil Adamczewski

Machine Learning, Generative Models, Diffusion

WROCŁAW 2025

Streszczenie

Synteza nowych widoków obiektów na podstawie pojedynczych obrazów pozostaje jednym z najbardziej wymagających problemów w widzeniu komputerowym. Istniejące metody oparte na modelach dyfuzyjnych borykają się z kompromisem między efektywnością obliczeniową a jakością wyników. Niniejsza praca wprowadza nowatorskie podejście łączące Feature-wise Linear Modulation (FiLM) do warunkowania parametrami kamery oraz równoległe adaptery atencji krzyżowej dla warunkowania widokiem referencyjnym. Proponowana metoda trenuje jedynie $585M$ z $2.9B$ parametrów (20%) przy zachowaniu porównywalnej wydajności z pełnym dostrajaniem modelu, osiągając 4-krotnie szybsze uczenie. Walidacja na zbiorach danych ObjaverseXL [5] i Google Scanned Objects [7] wykazuje korzyści w zakresie efektywności obliczeniowej w porównaniu z najlepszymi metodami Zero123++ [19] i MVAdapter [14], choć z niższą jakością wyników wynikającą z ograniczonej skali danych treningowych.

Abstract

Novel view synthesis of objects from single images remains one of the most challenging problems in computer vision. Existing diffusion-based methods face a trade-off between computational efficiency and result quality. This work introduces a novel approach combining Feature-wise Linear Modulation (FiLM) for camera parameter conditioning with parallel cross-attention adapters for visual conditioning. The proposed method trains only $585M$ of $2.9B$ parameters (20%) while achieving comparable performance to full model fine-tuning, with $4\times$ faster training. Validation on ObjaverseXL [5] and Google Scanned Objects [7] datasets demonstrates computational efficiency benefits compared to state-of-the-art Zero123++ [19] and MVAdapter [14] methods, though with performance gaps attributable to constrained training data scale.

Contents

1. Introduction

1.1. Motivation

Novel view synthesis from single images remains one of computer vision's most challenging problems. While diffusion models have revolutionized 2D image generation, adapting them for 3D-aware synthesis poses fundamental challenges: how to effectively encode geometric transformations without disrupting pre-trained knowledge, and how to balance computational efficiency with model expressiveness.

Current approaches suffer from critical limitations. Methods like Zero-1-to-3 [19] use simple pose vectors that provide insufficient geometric detail for complex viewpoint changes. Advanced techniques like CAT3D [9] and MV-Adapter [14] rely on complex raymap representations that are likely to introduce visual artifacts and struggle with reflective materials. Meanwhile, most of the existing methods require full fine-tuning large diffusion models which is computationally prohibitive.

1.2. Contributions

In this work, I introduce a novel approach that combines Feature-wise Linear Modulation (FiLM) [23] for camera parameter conditioning with parallel image cross-attention adapters for visual conditioning. My key contributions are:

1. **Hybrid conditioning architecture**: A dual-stream approach combining global geometric modulation with selective visual attention, addressing fundamental trade-offs in novel view synthesis.

2. **Efficient adapter training**: Training only $585M$ of $2.9B$ parameters (20%) while achieving comparable performance to full fine-tuning, with $4\times$ faster training.

3. **Feature-wise modulation for camera conditioning**: First application of FiLM to camera parameter conditioning on diffusion models, providing direct geometric control.

4. **Systematic evaluation**: Comprehensive ablation studies and comparison with state-of-the-art methods, demonstrating effectiveness across metrics and datasets.

The proposed method demonstrates significant computational efficiency gains compared to Zero123++ [19] and MVAdapter [14], though with performance limitations due to constrained training data scale.

1.3. Research Questions

This work addresses four specific research questions:

- Can Feature-wise Linear Modulation provide an effective alternative to complex raymap representations for camera parameter encoding?
- Can a hybrid conditioning strategy combining visual and geometric information achieve superior results compared to existing single-modal approaches?
- What is the optimal balance between training efficiency and model expressiveness in adapter-based approaches for novel view synthesis?
- How can we effectively condition the diffusion process to generate novel views of objects from a single reference image?

1.4. Structure

Chapter 2 reviews related work in 3D reconstruction, diffusion models, and novel view synthesis. Chapter 3 details the proposed method, including theoretical justification and architectural design. Chapter 4 describes data preparation with emphasis on lighting consistency between rendered images. Chapter 5 presents experimental evaluation including ablation studies and comparisons with state-of-the-art methods. Chapter 6 concludes with limitations, and future work.

2. Related Work

In this chapter, I provide a comprehensive review of existing approaches relevant to novel view synthesis from single or few input images. The chapter begins by introducing the fundamental task of Novel View Synthesis (NVS) and its significance in computer vision and 3D understanding. We then examine traditional 3D reconstruction techniques (Section 2.1) and their inherent limitations, particularly for sparse-input scenarios, which motivates the exploration of generative methods. Subsequently, the discussion shifts to the foundational principles of Deep Generative Models for Image Synthesis, with a focus on diffusion models (Section 2.2).

Building on this foundation, we explore various methods for Conditioning Diffusion Models for Enhanced Control and Adaptation to New Tasks (Section 2.3), including crucial techniques for camera parameter encoding and lightweight adaptation mechanisms. The core of the chapter then examines state-of-the-art Diffusion-based Novel View Synthesis techniques (Section 2.4), covering both single-image novel view synthesis and architectures for coherent multi-view generation.

2.1. Traditional 3D Reconstruction Approaches

Before examining generative approaches to novel view synthesis, it is essential to understand the limitations of traditional 3D reconstruction methods that motivate the shift toward diffusion-based solutions.

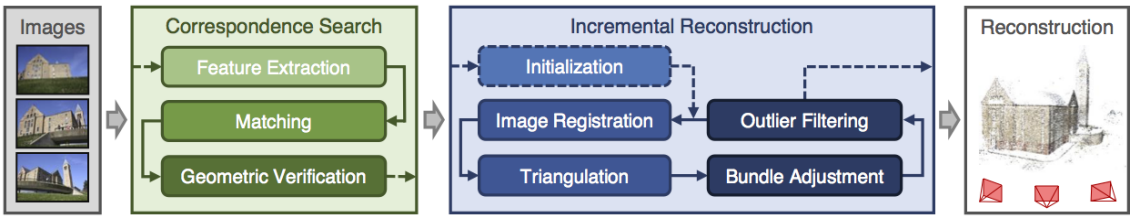

Figure 2.1: COLMAP pipeline

Structure from Motion (SfM) represents the classical approach to 3D reconstruction from images. Methods like COLMAP [28] demonstrate remarkable success in reconstructing 3D scenes from dense image collections through a two-stage pipeline 2.1: correspondence search to identify and match features across images, followed by incremental reconstruction to estimate camera poses and triangulate 3D points.

The success of SfM relies heavily on several key components: robust feature detection (typically using SIFT [18] or ORB [27]), accurate feature matching across viewpoints, and bundle adjustment to refine camera poses and 3D points. Dense reconstruction methods like Multi-View Stereo (MVS) can then generate detailed 3D models from the sparse SfM output.

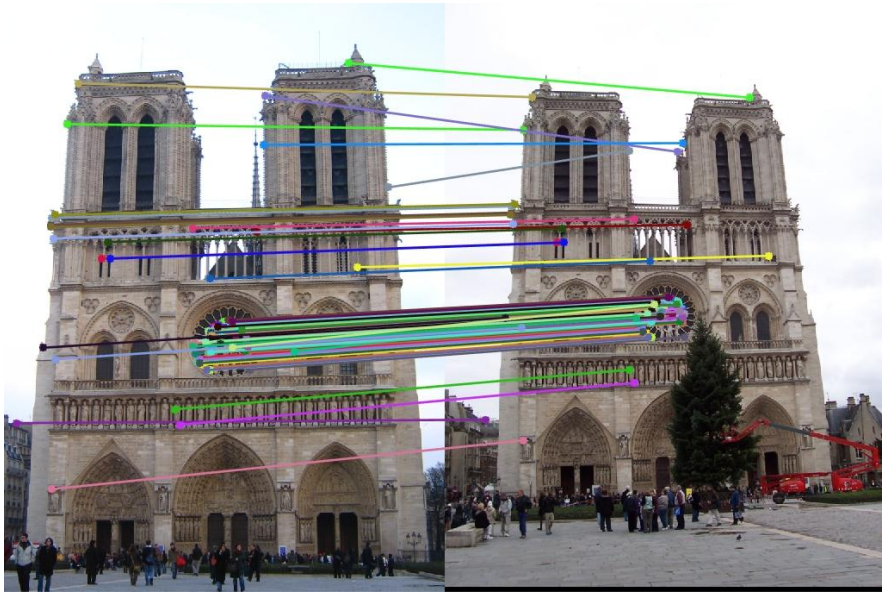

Figure 2.2: Feature matching across multiple views - a fundamental requirement for traditional SfM approaches

More recent learning-based approaches, including Neural Radiance Fields (NeRF) [21], have shown remarkable results for novel view synthesis. NeRF represents scenes as continuous 5D functions encoded in neural networks, taking camera poses from SfM as input and using volume rendering to synthesize photorealistic novel views with impressive detail and view consistency.

2.1.1. Fundamental Limitations for Sparse Input Scenarios

Traditional 3D reconstruction methods face critical limitations when applied to sparse input scenarios - particularly the extreme case of single-image novel view synthesis that forms the focus of this thesis:

1. **Insufficient correspondences**: SfM requires reliable feature matches across multiple views. With only one or very few images, establishing robust correspondences becomes impossible.

2. **Geometric ambiguity**: Single images provide incomplete depth information and cannot resolve occlusions, leading to fundamental ambiguities in 3D structure.

3. **View-dependent effects**: Materials with specular reflections or varying appearance cannot be accurately modeled without observing them from multiple known viewpoints.

4. **Scale and pose estimation**: Without multiple views, determining absolute scale and object pose becomes severely under-constrained.

These limitations have motivated the development of generative approaches to novel view synthesis, particularly using diffusion models trained on large datasets of 3D assets. Instead

of explicitly reconstructing geometry through correspondences, these methods leverage the power of deep learning to learn and synthesize plausible views from learned priors about object appearance and structure.

Modern 3D generation pipelines, such as InstantMesh [34], exemplify this paradigm shift by using diffusion models for multi-view generation as a first step, followed by 3D reconstruction from the synthesized views.

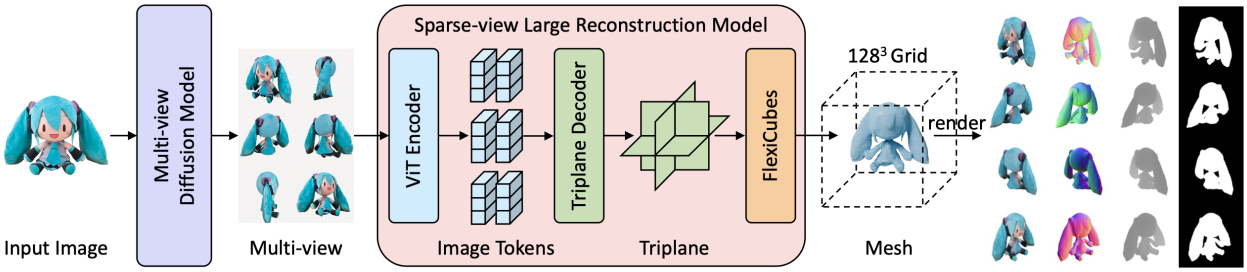

Figure 2.3: InstantMesh pipeline showing novel view synthesis as the foundation for 3D reconstruction

This transition from geometric reconstruction to generative synthesis motivates the focus of this thesis on developing efficient and effective diffusion-based approaches for novel view synthesis.

2.2. Deep Generative Models for Image Synthesis

The development of deep learning methods has led to remarkable progress in generative modeling, enabling the synthesis of highly realistic and diverse images. Among the various approaches, Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Diffusion Models have emerged as the most prominent. While GANs and VAEs have laid significant groundwork and continue to be influential, this section will primarily focus on diffusion models, given their state-of-the-art performance in image quality and controllability, and their direct relevance to the multi-view synthesis tasks explored in this thesis.

2.2.1. Generative Adversarial Networks and Variational Autoencoders

Generative Adversarial Networks (GANs) [10] consist of two neural networks, a generator and a discriminator, trained in a competitive setting. The generator aims to produce realistic images, while the discriminator tries to distinguish between real images from the training dataset and fake images produced by the generator. Through this adversarial process, the generator learns to create increasingly plausible images. GANs are known for generating sharp images but often suffer from training instability.

Variational Autoencoders (VAEs) [16] are another class of generative models that learn a probabilistic mapping from a high-dimensional data space (e.g., images) to a lower-dimensional latent space, and then back to the data space. A VAE consists of an encoder that compresses

the input data into a latent representation (typically a mean and variance defining a Gaussian distribution) and a decoder that reconstructs the data from samples drawn from this latent distribution. They are trained to maximize the evidence lower bound (ELBO), which involves a reconstruction loss and a regularization term (KL divergence) that encourages the latent space to be smooth and well-behaved. VAEs generally offer more stable training than GANs and can learn a meaningful latent space, but often produce slightly blurrier images. VAEs play a crucial role in the architecture of Latent Diffusion Models.

2.2.2. Diffusion Models

Diffusion models have recently become the dominant paradigm in high-fidelity image generation. They are inspired by non-equilibrium thermodynamics, specifically diffusion processes.

The Diffusion Process

The core idea behind diffusion models involves two processes: a forward (or diffusion) process and a reverse (or denoising) process. In the **forward process**, a known image x_0 from the dataset is gradually perturbed by adding small amounts of Gaussian noise over a sequence of T steps. This process progressively corrupts the image until, at step T, it becomes indistinguishable from pure isotropic Gaussian noise. The parameters of this noising process are fixed.

The **reverse process** aims to learn to reverse this noising. Starting from pure noise (equivalent to x_T), a neural network is trained to gradually denoise the signal, step-by-step, eventually producing a realistic image (an approximation of x_0). This learned denoising process is what allows the model to generate new images. Figure 2.4 illustrates this concept.

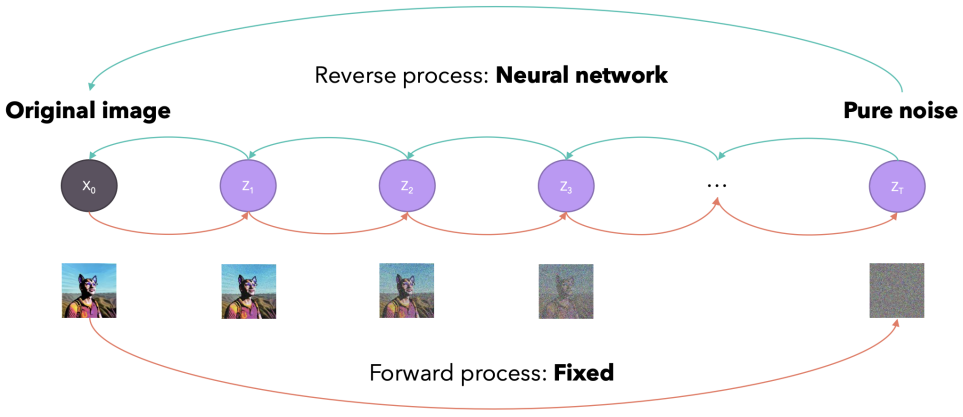

Figure 2.4: The forward (noising) and reverse (denoising/generation) stages of a diffusion model. The forward process gradually adds noise to an image until it becomes pure noise. The reverse process learned by a neural net to denoise, step-by-step, to generate an image from noise.

The U-Net Architecture

The neural network responsible for predicting the noise (or the denoised image) at each step in the reverse process is typically a U-Net architecture [26]. The U-Net, originally developed for biomedical image segmentation, features an encoder-decoder structure with skip (residual) connections. The encoder path progressively downsamples the input, capturing contextual information, while the decoder path progressively upsamples, localizing information. The skip connections concatenate features from the encoder to corresponding layers in the decoder, allowing the network to combine high-level semantic information with low-level detail, which is crucial for accurately predicting the noise and preserving image fidelity during the denoising steps.

Training Objective and Loss Function

The training objective of the diffusion model is to learn the conditional probability distribution $p_\theta(x_{t-1}|x_t)$, which represents the probability of the previous, less noisy state x_{t-1} given the current noisy state x_t. In practice, this is often simplified to training the U-Net to predict the noise ϵ that was added to an image x_0 to produce x_t at a given timestep t. The loss function is typically the Mean Squared Error (MSE) between the true added noise ϵ and the noise $\epsilon_\theta(x_t, t)$ predicted by the U-Net:

$$L = \mathbb{E}_{t\sim[1,T],x_0\sim q(x_0),\epsilon\sim\mathcal{N}(0,I)}[\|\epsilon - \epsilon_\theta(x_t, t)\|^2]$$

where $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, and $\bar{\alpha}_t$ are parameters from the noise schedule.

Conditional Generation: Text-to-Image Synthesis

To guide the image generation process, diffusion models can be conditioned on various inputs, most notably text prompts. This is the foundation of text-to-image synthesis. A crucial component for text conditioning is a powerful text encoder that can convert textual descriptions into rich numerical representations (embeddings). Contrastive Language-Image Pre-training (CLIP) [24] is widely used for this purpose. CLIP is trained on a massive dataset of image-text pairs to learn a shared embedding space where semantically similar images and texts are close together.

The text embeddings from CLIP are then integrated into the U-Net, typically using cross-attention mechanisms. In these layers, the image representation at an intermediate layer of the U-Net queries the text embedding, allowing the model to align parts of the image with relevant words or phrases in the prompt. This enables fine-grained control over the generated image content based on the textual input.

2.2.3. Latent Diffusion Models (LDMs)

While standard diffusion models operate directly in the pixel space of images, this can be computationally very expensive, especially for high-resolution images, as the U-Net needs to process large tensors. Latent Diffusion Models (LDMs) [25], such as Stable Diffusion, address this challenge by performing the diffusion and denoising process in a lower-dimensional latent space.

LDMs employ a pre-trained autoencoder, typically a VAE. The VAE's encoder first compresses a high-resolution image from pixel space into a compact latent representation. The diffusion process (both forward and reverse) then occurs entirely within this latent space. Once the reverse denoising process generates a target latent representation from noise, the VAE's decoder maps this latent representation back into the high-resolution pixel space to produce the final image. Figure 2.5 depicts the architecture of an LDM.

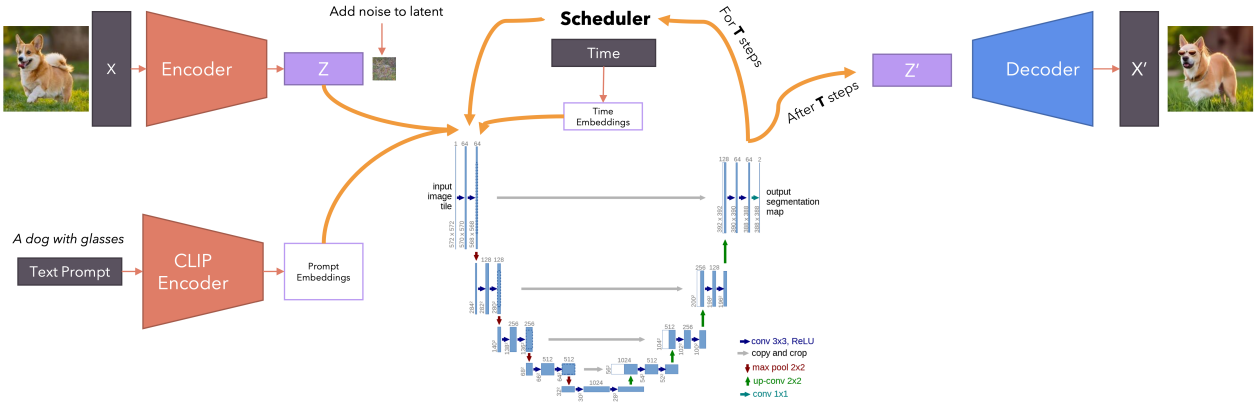

Figure 2.5: Architecture of a Latent Diffusion Model (LDM). An image is first encoded into a latent space by a VAE encoder. The diffusion process (noising and denoising via U-Net) occurs in this latent space. The generated latent is then decoded back to pixel space by the VAE decoder. Conditioning, such as text embeddings from CLIP, is incorporated into the U-Net.

By working in a compressed latent space, LDMs significantly reduce the computational burden during training and inference compared to pixel-space diffusion models. This makes it feasible to train powerful models on massive datasets and generate high-resolution images more efficiently, without a substantial loss in quality. The VAE ensures that the latent space is perceptually equivalent to the pixel space, and the diffusion model learns to generate high-quality latents within this space.

2.3. Conditioning Diffusion Models for Enhanced Control and New Tasks

While textual prompts provide a powerful and intuitive way to guide diffusion models, the generation process can be conditioned on a much wider array of signals. This opens up possibilities for more fine-grained control over the output and enables new applications beyond text-to-image synthesis. This section explores several advanced conditioning mechanisms, focusing on image-based conditioning for detailed content and style control, and specialized techniques for incorporating geometric information, such as camera parameters, to enrich 2D diffusion models with 3D awareness.

2.3.1. Image-based Conditioning for Fine-Grained Control

Conditioning on reference images allows for precise control over spatial layout, object appearance, color consistency, or artistic style, going beyond what is often achievable with text alone.

ControlNet

ControlNet [37] introduces a method to add diverse spatial conditioning to pre-trained text-to-image diffusion models. Instead of fine-tuning the original large model, ControlNet involves training smaller, task-specific auxiliary networks that are attached to the frozen U-Net backbone of the diffusion model, typically to its encoder blocks. These auxiliary networks take an input conditioning image (e.g., an edge map, human pose skeleton, depth map, or segmentation map) and produce feature maps that are then added to the corresponding features of the main U-Net. This approach allows the pre-trained model to be guided by the spatial information in the conditioning image while retaining its vast generative capabilities learned from large datasets. The original U-Net weights are locked, making ControlNet modules efficient to train and portable. Figure 2.6 provides a conceptual overview.

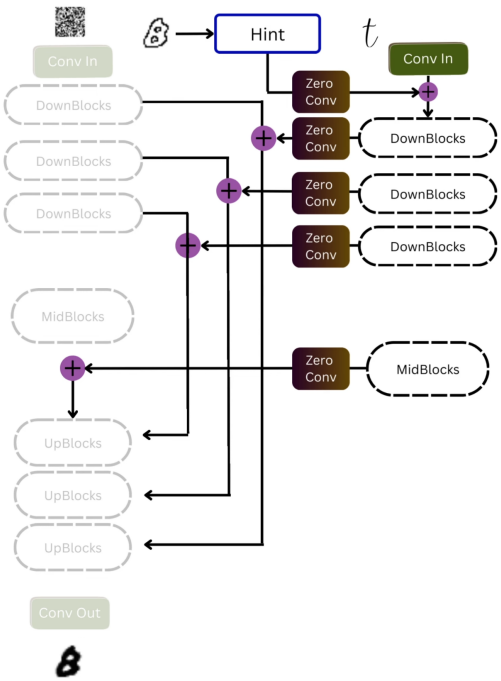

Figure 2.6: Conceptual overview of ControlNet, showing how an additional control module processes an input condition (e.g., edges, pose) and injects guidance into a pre-trained diffusion model. (Adapted from Zhang et al. [37])

IP-Adapter

IP-Adapter (Image Prompt Adapter) [36] offers an effective way to condition diffusion models using an image prompt, allowing the model to generate images that align with the

content or style of the reference image. It achieves this by introducing a lightweight adapter module that can be plugged into a pre-trained text-to-image model. The core idea is to decouple the cross-attention mechanisms used for text and image features. Image features, extracted by an image encoder (like CLIP [24] image encoder), are fed into new cross-attention layers that work in parallel with the original text cross-attention layers. This allows the model to draw information from both text and image prompts simultaneously or prioritize one over the other. IP-Adapter is efficient as it avoids fine-tuning the large base model and only trains the adapter parameters, making it a versatile tool for tasks like style transfer or subject-driven generation. Figure 2.7 illustrates this concept.

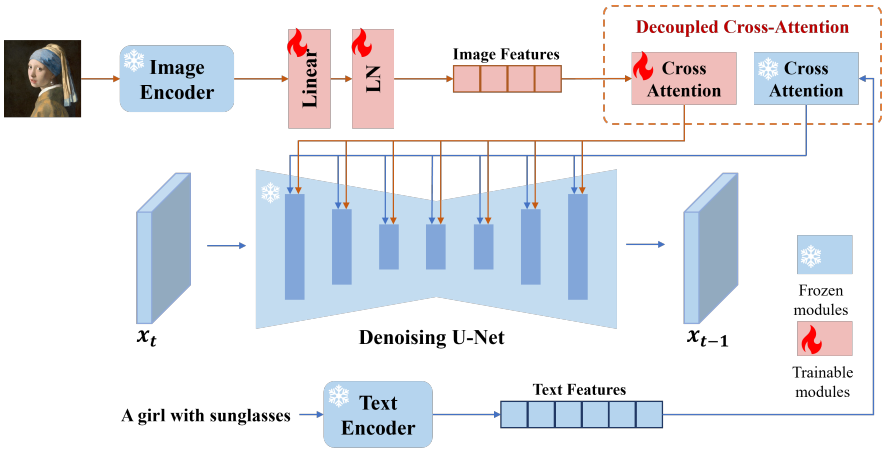

Figure 2.7: Conceptual illustration of the IP-Adapter architecture, showing how image features extracted by an image encoder are integrated into the U-Net via dedicated cross-attention layers to guide the generation process. (Image based on the IP-Adapter concept)

CatVTON and UNet Feature Conditioning

A particularly relevant approach for image conditioning, especially for tasks requiring detailed transfer of appearance or garment characteristics, is found in methods like CatVTON [3]. The key insight in such methods is to leverage the rich, hierarchical features learned by the denoising U-Net itself. When provided with a reference image (e.g., an image of a specific garment), features can be extracted from various layers of the U-Net as it processes this reference. These extracted features, which encode information about texture, shape, and style at multiple scales, are then used to condition the generation of a new image (e.g., the same garment, but worn by a person). This conditioning can be achieved by injecting these features into the corresponding layers of the U-Net during the denoising process of the target image, often using attention mechanisms or direct feature fusion. This technique allows for a powerful form of image-based control that is highly attuned to the diffusion model's internal representations, facilitating tasks like virtual try-on with impressive fidelity by directly utilizing the U-Net's understanding of visual elements.

2.3.2. Modulating Network Activations

Feature-wise Linear Modulation (FiLM) [23] is a general and effective technique for conditioning neural network activations. Instead of directly concatenating conditioning information or using complex gating mechanisms, FiLM applies a simple affine transformation to feature maps based on a conditioning input. Given a feature map h from a layer in a neural network, and a conditioning vector c, a FiLM generator (typically a small neural network) produces a scale parameter γ and a shift parameter β from c. These parameters then modulate h as follows: $FiLM(h) = \gamma \cdot h + \beta$. This allows the conditioning information c to dynamically influence the behavior of the network layer by layer. FiLM is lightweight and can be integrated into various architectures to allow secondary inputs to control the processing of a primary input. Figure 2.8 illustrates the internals of a FiLM layer.

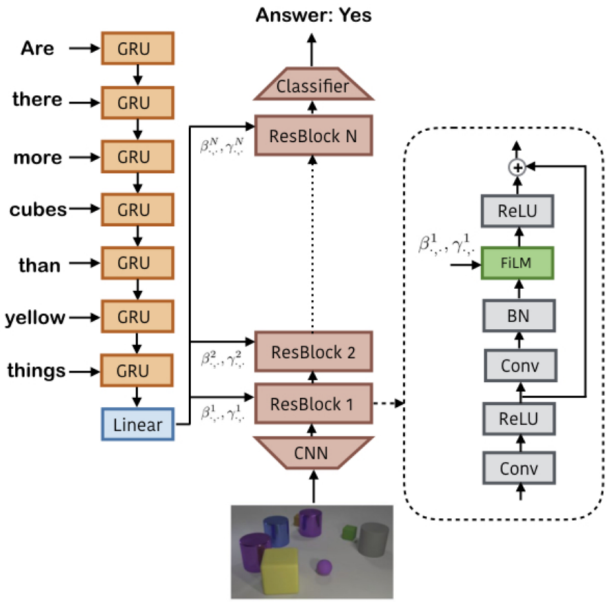

Figure 2.8: The Feature-wise Linear Modulation (FiLM) layer. A conditioning input is processed by a FiLM generator to produce scale (γ) and shift (β) parameters, which then modulate the feature maps (h) of a main network. (Adapted from Perez et al. [23])

This approach enabled image-based models to effectively integrate and act upon conditioning information from outside the image domain. For instance, by modulating visual features based on encoded text tokens, it empowered (at the time) state-of-the-art image question answering models to ground their visual processing in textual context.

FiLMed-UNet

The FiLM technique has also been effectively applied to U-Net architectures in the medical imaging domain, notably in the FiLMed-UNet paper by Lemay et al. [17]. Their work focused on enhancing medical image segmentation by incorporating various forms of metadata as conditioning signals. FiLMed-UNet utilized patient-specific information, such as tumor type or the target organ for segmentation.

The core idea was to make the U-Net's segmentation process adaptable and more informed by this external metadata. The metadata (e.g., one-hot encoded tumor type) was passed through a FiLM generator network, which produced scale (γ) and shift (β) parameters. These parameters then modulated the feature maps at different layers within the U-Net. This allowed the network to tailor its feature extraction and segmentation logic based on the provided metadata. For example, knowing the tumor type allowed the model to leverage type-specific visual characteristics, leading to improved segmentation accuracy. Furthermore, by conditioning on the desired output class (e.g., "segment kidney"), FiLMed-UNet demonstrated robustness in multi-task learning scenarios, even with missing labels for some tasks in the training data, and showed improved performance with limited annotations. While this application of FiLM is distinct from encoding camera parameters for 3D-aware synthesis, it highlights the versatility of FiLM in allowing neural networks to integrate and utilize diverse conditioning information to adapt their behavior for specialized tasks. Figure 2.9 shows a conceptual diagram.

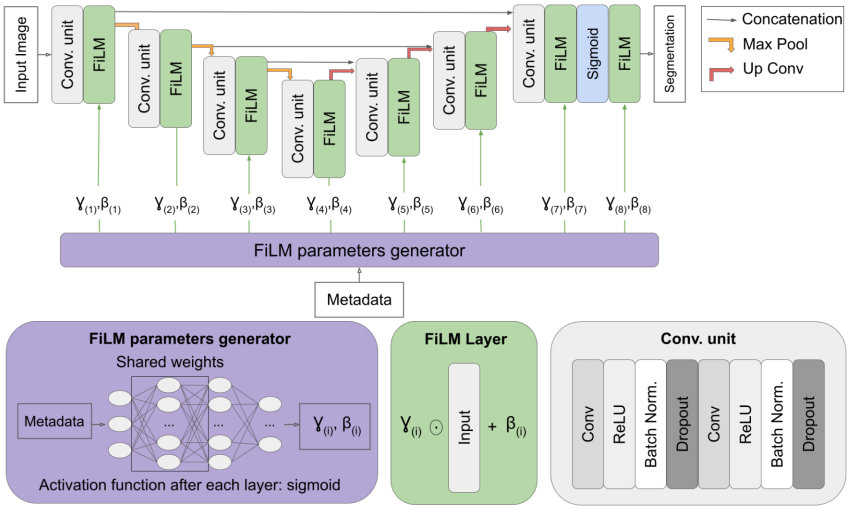

Figure 2.9: Conceptual diagram of a FiLMed-UNet. The metadata about patients are processed by a FiLM generator network, which outputs scale and shift parameters to modulate activations at various layers within the U-Net architecture, making the denoising process viewpoint-aware.

2.3.3. Camera Parameter Encoding for 3D Awareness

Making 2D diffusion models inherently 3D-aware is crucial for tasks like multi-view image generation. A key aspect of this is encoding camera parameters (Rotation R, Translation t) to guide the image generation process. Various methods focus on integrating this geometric information directly into the model's input or intermediate representations.

Pixel-Level Geometric Conditioning: Ray-based Approaches

This category of methods aims to provide detailed, per-pixel geometric information derived from camera parameters directly to the diffusion model. A prominent technique is the use of

a *raymap*, as demonstrated by Watson et al. [33] in 3DiM and also utilized in CAT3D [9]. The raymap is a tensor with the same spatial dimensions as the image or latent representation. At each (u,v) coordinate, it stores the 3D origin and 3D direction of the camera ray passing through that pixel. In CAT3D, these rays are computed relative to the camera pose of the first conditional image. To help the network learn high-frequency details associated with viewpoint changes, these ray origins and directions are often further processed using positional encodings. The resulting positionally encoded raymap can be concatenated as an additional input channel to the U-Net, alongside the noisy image or latent representation [9, 33]. This allows the model to learn a direct, spatially varying mapping from ray geometry to the target pixel output.

Global Camera Pose Conditioning

Alongside per-pixel data, global camera pose conditioning methods provide a more compact, holistic representation of the camera's viewpoint or transformation. One approach involves *vectorized transformation embeddings*. For instance, Zero-1-to-3 [19] encodes the relative camera viewpoint transformation $T = [R|t]$ between an input view and a target view as a 4x4 matrix. This matrix is flattened into a 16-dimensional vector, which is then projected by a small Multi-Layer Perceptron (MLP) and added to the timestep embedding. This globally influences the diffusion process by providing context about the desired viewpoint change. Another technique employs *decomposed pose vectors for attention mechanisms*. MV-Adapter [14] represents each camera view using a 12-dimensional vector, which includes 3D camera coordinates and a 9D flattened rotation matrix. These vectors are processed by an MLP and then fed into multi-view attention layers as key and value components. This allows the model to explicitly attend to pose information when correlating features across different views, aiding in the generation of coherent multi-view imagery.

2.4. Diffusion-based Novel View Synthesis

Building upon the camera parameter encoding techniques discussed in Section 2.3.3, this section examines specific diffusion-based models and architectures designed for novel view synthesis from single or multiple input images. These methods aim to leverage the generative power of diffusion models to synthesize novel perspectives while maintaining consistency with input views and underlying 3D structure.

2.4.1. Zero-1-to-3: Pioneering Single-Image Novel View Synthesis

Zero-1-to-3 [19] represents a foundational contribution to diffusion-based Novel View Synthesis, demonstrating that large-scale pre-trained diffusion models contain rich geometric priors that can be leveraged for view synthesis tasks. Beyond the methodological contribution, Zero-1-to-3 introduced the use of the Objaverse dataset [5] for training diffusion-based novel view synthesis models, which has become a standard training resource for subsequent work in this field. The method fine-tunes Stable Diffusion to condition on both a reference image and relative camera transformations, enabling zero-shot novel view synthesis from a single RGB image.

Technical Approach: Zero-1-to-3 employs a hybrid conditioning mechanism combining high-level semantic information through CLIP embeddings and low-level detail preservation through direct image concatenation. The relative camera transformation (R, T) is encoded as a compact vector and integrated with the CLIP embedding of the input image to form a "posed CLIP" embedding for cross-attention conditioning.

Key Contributions: The work demonstrates that Internet-scale pre-training enables diffusion models to learn geometric relationships between viewpoints, even when trained only on 2D images. This insight opened the door for subsequent work in diffusion-based 3D generation.

Limitations: While groundbreaking, Zero-1-to-3 can struggle with maintaining geometric consistency across widely varying viewpoints, particularly for complex objects with intricate geometry or materials.

2.4.2. MVDream: Multi-View Consistent Generation

MVDream extends the paradigm established by Zero-1-to-3 to generate multiple consistent views simultaneously from text prompts. Rather than generating views independently, MVDream introduces multi-view attention mechanisms that enable reasoning about 3D consistency during the generation process.

Architecture: MVDream modifies the self-attention layers of a pre-trained diffusion model to operate across multiple views simultaneously. Features from different viewpoints are concatenated and processed through extended attention mechanisms, allowing the model to enforce consistency constraints between views.

Training Strategy: The model is trained on both 2D image datasets and rendered multi-view data from 3D assets, learning to balance high-quality image generation with 3D consistency. A canonical camera coordinate system is established where the default view corresponds to the object's front-facing orientation.

Impact: MVDream has become a foundational model for many subsequent 3D generation pipelines, particularly those using Score Distillation Sampling (SDS) for optimizing neural radiance fields.

2.4.3. ImageDream: Image-Conditioned Multi-View Generation

Building on MVDream's multi-view attention framework, ImageDream [31] introduces image-prompt conditioning for multi-view generation. This addresses the common requirement of generating consistent views from a single reference image rather than just text descriptions.

Multi-Level Image Conditioning: ImageDream integrates image features at three levels:

- **Global level**: CLIP global embeddings for high-level semantic understanding
- **Local level**: CLIP hidden features for intermediate spatial information
- **Pixel level**: VAE-encoded image latents concatenated as an additional frame for detailed appearance transfer

Technical Innovation: The pixel-level controller treats the input image as an additional view in the multi-view attention mechanism, enabling 3D self-attention across all five frames (four target views plus the reference image).

2.4.4. CAT3D: Large-Scale Consistent View Generation

CAT3D [9] presents a multi-view diffusion model designed to simulate realistic capture processes by generating large sets of consistent novel views (80+ views for single-image input, up to 960 for few-view scenarios).

Architecture Design: CAT3D employs a video-latent-diffusion-like architecture with 3D self-attention (2D spatial + 1D temporal across views). The model is initialized from pre-trained latent diffusion models to leverage existing image priors.

Camera Conditioning: Following the raymap approach from 3DiM [33], CAT3D computes ray origins and directions relative to the first conditional image's camera pose. These are positionally encoded and concatenated channel-wise to the latent representations.

Scalable Generation: For large view counts, CAT3D employs a hierarchical generation strategy, first sampling anchor views autoregressively, then generating remaining views conditioned on the anchors.

2.4.5. MV-Adapter: Efficient Adapter-Based Multi-View Generation

MV-Adapter [14] introduces the first adapter-based solution for multi-view generation, addressing computational efficiency concerns while preserving the quality and capabilities of large pre-trained models.

Decoupled Attention Architecture: Rather than modifying existing self-attention layers, MV-Adapter duplicates these layers to create specialized multi-view attention modules. This preserves the original feature space while enabling efficient training of only the adapter parameters.

Parallel Organization: The duplicated attention layers are organized in parallel rather than serial architecture, allowing them to inherit pre-trained image priors effectively while learning geometric knowledge. Output projections are zero-initialized to prevent disruption of the original feature space.

Unified Conditioning: MV-Adapter supports both camera parameter conditioning (through raymaps) and geometric conditioning (through position and normal maps), enabling applications in both object generation and texture synthesis.

Adaptability: The adapter design allows integration with various T2I model derivatives, including personalized models, distilled models, and control networks, significantly expanding application possibilities.

2.4.6. Architectural Innovations and Technical Insights

The evolution of diffusion-based novel view synthesis reveals several key architectural patterns and insights:

Attention Mechanisms: The progression from single-view conditioning (Zero-1-to-3) to multi-view attention (MVDream) to adapter-based attention (MV-Adapter) demonstrates the importance of architectural choices in balancing quality, consistency, and efficiency.

Camera Representation: Various camera conditioning approaches have emerged, from simple pose vectors in Zero-1-to-3 to sophisticated raymap representations in CAT3D and MV-Adapter. The choice of representation significantly impacts the model's ability to generalize to diverse viewpoints.

Training Efficiency: The shift toward adapter-based approaches (MV-Adapter) addresses practical limitations of full model fine-tuning, enabling training on larger base models and higher resolutions while preserving pre-trained knowledge.

Multi-Scale Integration: Methods like ImageDream demonstrate the importance of multi-level feature integration for effective image conditioning, combining global semantic understanding with detailed appearance transfer.

Despite significant progress, current methods face ongoing challenges in geometric consistency for complex viewpoint changes, computational efficiency for high-resolution generation, and generalization to diverse object categories and lighting conditions. These limitations motivate the development of more efficient and robust approaches to diffusion-based Novel View Synthesis.

3. Proposed Method

In this chapter, I describe the proposed method for adapting a pre-trained 2D diffusion model to understand 3D geometry and synthesize novel views of objects. The approach focuses on conditioning the generative process on both a reference image and relative camera transformations.

3.1. Research Motivation and Gaps

Despite the significant progress in diffusion-based novel view synthesis, several critical limitations motivate the development of my approach:

1. **Training Efficiency vs. Model Expressiveness**: Existing approaches face a fundamental trade-off between computational efficiency and model expressiveness. While MV-Adapter [14] introduced adapter-based training, full fine-tuning remains computationally prohibitive for large models, yet extremely lightweight adapters may limit the model's ability to learn complex 3D relationships.

2. **Camera Conditioning Limitations**: Current methods rely on either simple pose vectors (Zero-1-to-3 [19]) that provide insufficient geometric detail for complex viewpoint changes, or complex raymap representations (CAT3D [9], MV-Adapter [14]) that struggle with reflective materials and often introduce visual artifacts disrupting realistic image synthesis.

3. **Feature Integration Strategies**: Most existing methods either use cross-attention for all conditioning (requiring significant architectural modifications) or global feature concatenation (limiting spatial awareness). There is a gap in exploring hybrid conditioning strategies that leverage both global geometric understanding and detailed visual feature transfer.

4. **Training Data Quality and Lighting Inconsistencies**: Standard Objaverse rendering pipelines employ randomized lighting setups that often result in harsh lighting, strong shadows, or underexposed scenes due to unfavorable light positioning. This inconsistency severely impacts the model's ability to learn reliable appearance and geometric relationships, directly affecting downstream 3D reconstruction applications. Chapter 4 details the approach to address this issue.

5. **Limited Viewpoint Generalization**: Existing methods trained on specific camera perspective sets often fail to generalize to novel viewpoint configurations, limiting their effectiveness for applications requiring flexible camera positioning and arbitrary viewpoint synthesis.

6. **Adaptation to Unseen Objects**: Current methods struggle to generalize beyond training data object categories and styles. This limitation becomes apparent when applying trained models to real-world objects with different materials, textures, or geometric properties, limiting practical applicability.

My Contributions: This work addresses these limitations through: (1) a balanced adapter architecture that maintains expressiveness while ensuring training efficiency, (2) a FiLM-based camera conditioning mechanism that enables direct geometric control through learnable camera parameter encoding, (3) a hybrid conditioning strategy combining parallel image cross-attention with network modulation, (4) carefully curated training data with consistent lighting conditions, (5) training on diverse viewpoint configurations as well as (6) diverse object categories to maximize generalization capabilities.

3.2. Theoretical Justification

The core innovation of this work lies in the application of Feature-wise Linear Modulation (FiLM) for camera parameter conditioning in diffusion models. This section provides theoretical justification for why this approach is fundamentally well-suited for novel view synthesis.

3.2.1. Geometric Transformations and Feature Space Modulation

Camera transformations between viewpoints can be decomposed into rotation matrix $\mathbf{R} \in SO(3)$ and translation vector $\mathbf{t} \in \mathbb{R}^3$, which define how 3D points transform between coordinate systems. In the context of novel view synthesis, we seek to transform feature representations from a source viewpoint to a target viewpoint.

The key insight is that FiLM's affine transformation $\gamma \odot \mathbf{h} + \beta$ provides a learnable approximation of geometric transformations in feature space. Specifically:

- **Scaling component** (γ): Can model perspective changes, depth-dependent scaling, and orientation-dependent feature emphasis that arise from rotational transformations
- **Shift component** (β): Can model translational offsets and view-dependent bias adjustments needed to account for position changes

3.2.2. Cross-Modal Conditioning

FiLM has been proven effective for cross-modal conditioning, where information from one modality (e.g., text) influences processing in another modality (e.g., images) [17, 23]. This work extends this framework by treating camera parameters as a distinct geometric modality.

The theoretical foundation rests on FiLM's ability to perform *conditional computation* - the same feature map $\mathbf{h}$ can be processed differently based on the conditioning signal. For camera conditioning:

$$\text{FiLM}(\mathbf{h}|\mathbf{R}, \mathbf{t}) = \gamma(\mathbf{R}, \mathbf{t}) \odot \mathbf{h} + \beta(\mathbf{R}, \mathbf{t})$$

where γ and β are learned functions of the camera parameters. This formulation allows the network to dynamically adjust its feature processing based on the target viewpoint, effectively implementing view-dependent feature transformations.

3.2.3. Hybrid Conditioning Strategy

The combination of FiLM-based camera conditioning with parallel image cross-attention is motivated by the complementary nature of these mechanisms:

- **FiLM conditioning**: Provides global, view-dependent modulation that affects how features are processed throughout the network
- **Image cross-attention**: Enables selective attention to relevant visual details from the reference image

This hybrid approach allows the model to both globally adapt its processing for the target viewpoint (via FiLM) and selectively transfer appropriate visual content (via attention), addressing the fundamental challenge of novel view synthesis: maintaining visual consistency while adapting to geometric changes.

3.3. Method Overview

My proposed method adapts Stable Diffusion 2.1 for multi-view novel view synthesis by introducing two specialized conditioning streams. The method leverages strong generative capabilities of the pre-trained 2D diffusion model while equipping it with 3D spatial understanding through minimal architectural modifications.

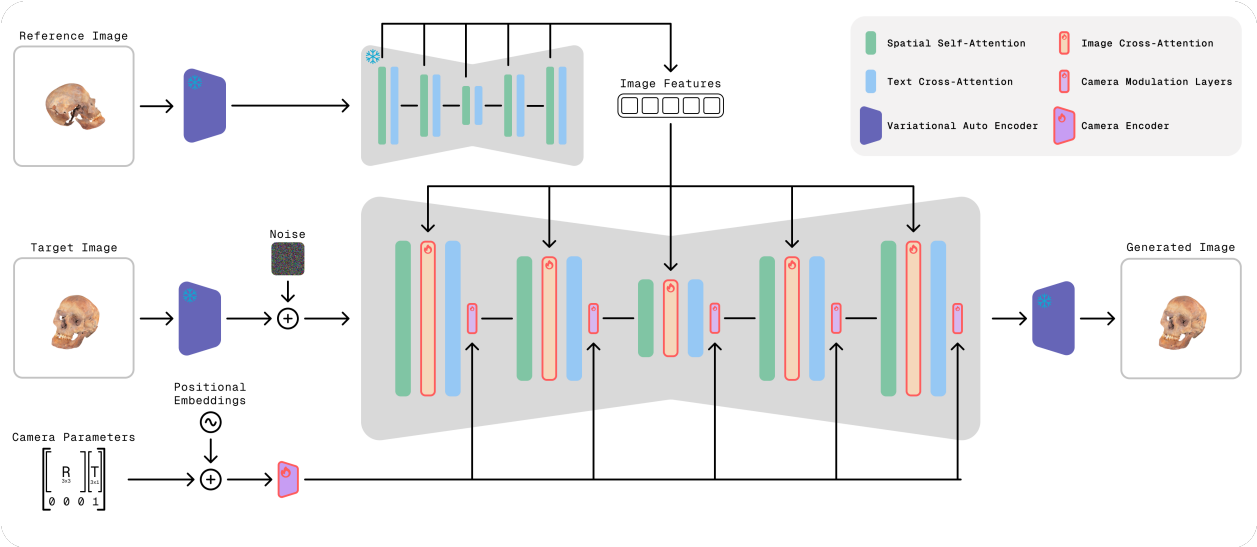

Figure 3.1: System diagram of the proposed multi-view diffusion model. It showcases the reference image encoding path, the camera parameter encoding path, and their integration into the main denoising U-Net.

The method adapts the Stable Diffusion 2.1 model, an inherently 2D generative framework, to comprehend 3D spatial relationships and generate images from novel viewpoints. The system achieves this by integrating two distinct conditioning streams into the diffusion model's U-Net architecture. The first stream provides visual context from a source (reference) image, while the second stream imparts geometric awareness through encoded camera parameters.

Figure 3.1 illustrates the overall architecture of the proposed system, highlighting the flow of information and the interplay between the original diffusion model components and the newly introduced conditioning modules.

Core Innovation: The key innovation lies in the hybrid conditioning approach that combines:

- **Visual conditioning**: Parallel image cross-attention adapters that preserve visual consistency
- **Geometric conditioning**: FiLM-based camera parameter modulation for spatial awareness
- **Efficiency**: Adapter-based training that keeps the backbone model frozen

3.4. Architecture

In my method I leverage the strong generative capabilities of a pre-trained 2D diffusion model and extend it for 3D-aware novel view synthesis. This is achieved through the introduction of specialized conditioning modules that inject visual and geometric information into the denoising U-Net.

3.4.1. Backbone: Stable Diffusion 2.1

The foundation of the proposed method is the Stable Diffusion 2.1 model. I chose this model for its robust text-to-image generation capabilities, clear architectural design, and its publicly available pre-trained weights, which encapsulate extensive understanding of visual concepts. The core component relevant to my modifications is its U-Net architecture [26], which performs the iterative denoising process.

In my method I keep several key components of the original model frozen to preserve their learned knowledge and ensure computational efficiency during training. These include the Variational Autoencoder (VAE) [16] used for encoding images into and decoding latents from the latent space, the text encoder CLIP [24] responsible for processing textual prompts, and the weights of the original U-Net when used within the Reference Image Encoder. I specifically designed the new conditioning modules and adapters to be trainable while maintaining the frozen backbone.

3.4.2. Reference Image Conditioning Stream

To enable the model to generate views that are visually consistent with a given input, a reference image conditioning stream is introduced. This approach is inspired by recent advances

in image-conditioned generation, particularly the feature extraction strategies employed in CatVTON [3] for garment transfer and the multi-view attention mechanisms developed in MV-Adapter [14]. This stream aims to capture and transfer the appearance, texture, color, and identity of the object depicted in the source image to the novel view generation process.

Source Image Feature Extraction

The process of extracting visual features from the source image through a dedicated ImageEncoder module leverages a frozen copy of the Stable Diffusion 2.1 U-Net to extract multi-scale attention features.

Feature Extraction Process: The method first encodes the source image into latent space using the VAE encoder. Then, the resulting latents are processed through a frozen copy of the Stable Diffusion 2.1 U-Net with timestep set to zero ($t = 0$), effectively performing feature extraction (rather than denoising). During this forward pass, the ImageEncoder registers forward hooks on attention layers throughout the U-Net architecture. This frozen U-Net copy is separate from the main denoising U-Net and serves purely for feature extraction, minimizing the number of trainable parameters. The hooks capture outputs from all the attention layers in the U-Net. These features encode visual information at different semantic levels, from low-level textures in early layers to high-level semantic concepts in deeper layers.

Image Cross-Attention Adapters

The visual features extracted by the ImageEncoder are injected into the main denoising U-Net using newly introduced, trainable adapter modules that serve as cross-attention layers. These processors are designed to be lightweight and integrated into each corresponding block of the denoising U-Net.

Parallel Architecture Design: A key architectural choice is the parallel integration of these image cross-attention adapters, inspired by the adapter design philosophy of MV-Adapter [14]. Instead of serially passing information through the original attention layers and then the new adapters, the adapters operate in parallel to the U-Net's existing self-attention and text-cross-attention mechanisms. This design choice preserves the original feature space while enabling efficient learning of image-conditioned representations, similar to the approach used in IP-Adapter [36] for image prompt conditioning.

Attention Mechanism: The extracted image features serve as the key ($\mathbf{K}_{img}$) and value ($\mathbf{V}_{img}$) for these new cross-attention layers, while the U-Net's intermediate hidden states act as the query ($\mathbf{Q}$). The cross-attention operation is computed using scaled dot-product attention:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}_{img}, \mathbf{V}_{img}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}_{img}^T}{\sqrt{d_k}}\right)\mathbf{V}_{img}$$

The output is scaled by a hyperparameter - reference scale factor and added to the original attention output:

$$\mathbf{h}_{out} = \mathbf{h}_{original} + \text{ref\_scale\_val} \cdot \text{Attention}(\mathbf{Q}, \mathbf{K}_{img}, \mathbf{V}_{img})$$

Efficiency Considerations: The adapter-based approach trains only 585M parameters out of the total 2.9B parameters of the complete model (including U-Net, VAE, text encoder, and adapters), ensuring computational efficiency while maintaining expressive power.

3.4.3. Camera Parameter Conditioning Stream

To ensure geometric consistency and enable precise control over the viewpoint of the generated image, the method employs a camera parameter conditioning stream. This stream encodes the relative transformation between the source and target camera poses and uses this information to modulate the behavior of the denoising U-Net.

Camera Pose Encoding

The CameraEncoder module processes geometric information by computing and encoding the relative transformation between source and target camera poses.

Relative Transformation Computation: Following the relative pose encoding strategy established in Zero-1-to-3 [19], given source camera pose $\mathbf{C}_s$ and target camera pose $\mathbf{C}_t$ (both 4×4 homogeneous transformation matrices), the method first extracts rotation matrices $\mathbf{R}_s, \mathbf{R}_t \in \mathbb{R}^{3\times3}$ and translation vectors $\mathbf{t}_s, \mathbf{t}_t \in \mathbb{R}^3$. The relative transformation is computed as:

$$\mathbf{R}_{rel} = \mathbf{R}_t \mathbf{R}_s^T$$

$$\mathbf{t}_{rel} = \mathbf{t}_t - \mathbf{R}_{rel}\mathbf{t}_s$$

Dual-Stream Encoding: The method processes the relative rotation and translation through separate encoding streams.

Rotation Encoding: The 3×3 rotation matrix are flattened to a 9-dimensional vector and processed through a dedicated MLP:

$$\mathbf{E}_{rot} = \text{MLP}_{rot}(\text{flatten}(\mathbf{R}_{rel}))$$

Translation Encoding: The translation vector $\mathbf{t}_{rel}$ is first positionally encoded to enrich its geometric representation, adapting the positional encoding technique from transformer architectures [30]:

$$\text{PE}(\mathbf{t}_{rel}) = [\sin(\omega_k \cdot \mathbf{t}_{rel}), \cos(\omega_k \cdot \mathbf{t}_{rel})]_{k=1}^{K}$$

where $\omega_k = \exp(\frac{k \log(\omega_{max})}{K})$ are logarithmically spaced frequencies. The positionally encoded translation is then processed through its own MLP:

$$\mathbf{E}_{trans} = \text{MLP}_{trans}(\text{PE}(\mathbf{t}_{rel}))$$

Final Projection: The rotation and translation embeddings are concatenated and projected to the final camera embedding dimension:

$$\mathbf{E}_{cam} = \text{MLP}_{final}([\mathbf{E}_{rot}; \mathbf{E}_{trans}])$$

where $\mathbf{E}_{cam} \in \mathbb{R}^{d_{cam}}$ represents the final camera embedding with dimension $d_{cam} = 2048$.

The MLP details are provided in the next chapter (Chapter 5).

FiLM-based Network Modulation

The high-dimensional camera embedding produced by the CameraEncoder is used to modulate the activations within the main denoising U-Net via Feature-wise Linear Modulation (FiLM) layers [23]. This approach is motivated by the success of FiLM in cross-modal conditioning tasks, particularly its application in FiLMed-UNet [17] for medical image segmentation with metadata conditioning. As detailed in Chapter 2, FiLM provides an effective mechanism for conditioning neural network activations. This work applies this mechanism specifically to camera parameter conditioning in diffusion models, enabling camera pose information to dynamically influence feature representations throughout the network.

Modulator Architecture: For each U-Net block that requires camera conditioning, the method employs dedicated modulator networks that generate scale and shift parameters from the camera embedding:

$$[\boldsymbol{\gamma}; \boldsymbol{\beta}] = \text{MLP}_{mod}(\mathbf{E}_{cam})$$

where MLP_{mod} is a two-layer MLP that maps the camera embedding to concatenated scale and shift parameters.

FiLM Operation: For a given feature map $\mathbf{h} \in \mathbb{R}^{B \times C \times H \times W}$ in the U-Net, the FiLM layer applies:

$$\text{FiLM}(\mathbf{h}) = \boldsymbol{\gamma} \odot \mathbf{h} + \boldsymbol{\beta}$$

where $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^C$ are channel-wise parameters, and $\odot$ denotes element-wise multiplication broadcast across spatial dimensions.

Network Integration: FiLM modulations are applied through forward hooks registered on the outputs of all the U-Net down-sampling blocks, middle block, and up-sampling blocks. The hooks intercept the block outputs and apply the corresponding camera-conditioned modulation before passing the features to subsequent layers.

Initialization Strategy: The modulator networks are initialized with small weights and biases set such that the initial scale parameters have mean 0.5 and shift parameters have mean 0.0. This ensures stable training initialization where the camera conditioning gradually integrates without disrupting pre-trained feature representations.

3.4.4. The Integrated Conditioned U-Net

The MultiViewUNet module serves as the central component of the proposed architecture. It integrates the backbone Stable Diffusion U-Net with the two conditioning streams described above: the reference image conditioning via parallel Image Cross-Attention Adapters and the camera parameter conditioning via Feature-wise Linear Modulation (FiLM) layers.

During each step of the reverse diffusion process, the MultiViewUNet takes the noisy latent representation of the target image, the current timestep, text embeddings, the extracted reference image features, and the encoded target camera parameters. It then predicts the noise present in the noisy latent. The system diagram in Figure 3.1 provides a visual summary of this integrated data flow.

3.5. Model Training and Inference

The training process is designed to teach the MultiViewUNet to effectively utilize the visual and geometric conditioning information to predict the noise required to generate a target view from a source view and camera transformation.

3.5.1. Training Objective and Loss Functions

The primary training objective is to minimize the difference between the noise predicted by the MultiViewUNet and the actual noise that was added to the target image's latent representation. This is formulated as a Mean Squared Error (MSE) loss:

$$L_{noise} = \mathbb{E}_{x_0,\epsilon,t}[\|\epsilon - \epsilon_\theta(x_t, t, \mathbf{c}_{img}, \mathbf{c}_{cam}, \mathbf{c}_{text})\|^2]$$

where x_0 is the clean target image latent, ϵ is the sampled Gaussian noise, t is the timestep, x_t is the noisy latent at timestep t, and ϵ_θ is the noise predicted by the network conditioned on image features $\mathbf{c}_{img}$, camera embedding $\mathbf{c}_{cam}$, and text embedding $\mathbf{c}_{text}$. The noise scheduler is configured to predict the noise term ϵ.

To improve training stability and sample quality, especially at varying noise levels, the method employs an SNR-aware weighting strategy for the loss function, specifically the Min-SNR-γ approach [11]. The loss is weighted for each training instance by $\min(SNR_t, \gamma)/SNR_t$, where SNR_t is the signal-to-noise ratio at timestep t, and γ is a hyperparameter (set to 5.0) as the method authors recommend [11]. This weighting scheme effectively gives more importance to timesteps with lower SNR values, preventing the model from focusing excessively on high-SNR (low noise) timesteps. The DDPMScheduler is used as the noise schedule, further modified using an interpolated shift based on the SNR, with a shift scale factor (set to 6.0), which adapts the noise levels experienced during training.

While primary loss focuses on noise prediction, a set of auxiliary metrics are monitored during validation steps to provide a comprehensive assessment of image quality and consistency. For rapid validation during hyperparameter tuning and intermediate training checks, Perceptual Loss alongside SSIM, CLIP score, and FID are used. For final model evaluation and reporting, more rigorous metrics such as LPIPS [38], CLIP score [12], and Fréchet Inception Distance (FID [13, 29]), in addition to PSNR and SSIM are used. More details about the metrics are and validation process are provided in the next chapter (Chapter 5).

3.5.2. Training Data and Iteration

Each training iteration involves a batch of data samples. A single sample consists of:

- A source image.
- A target image (representing a different view of the same object).
- The relative camera transformation (rotation $\mathbf{R}$ and translation $\mathbf{t}$) from the source camera pose to the target camera pose.

- A textual prompt describing the object (leveraging the underlying text-to-image capabilities of Stable Diffusion).

The model is trained to predict the noise in the target image's latent, conditioned on the source image, the camera transformation, and the text prompt.

3.5.3. Optimization and Implementation Details

The trainable parameters of the model include the weights of the modules mentioned in Section 3.4. The core U-Net of Stable Diffusion 2.1, the VAE, and the text encoder are kept frozen during this fine-tuning phase.

Optimization is performed using the AdamW optimizer [20] with a learning rate of 1×10^{-5}. A cosine learning rate schedule with a warm-up period is employed to manage the learning rate dynamics over the training duration, which is set for a total of 25 epochs. Gradient clipping with a maximum norm of 1.0 is applied to prevent exploding gradients. Training is performed using a batch size of 6 per GPU. The model is trained on 4 GPUs (NVIDIA A100 40GB). The model is implemented using PyTorch and PyTorch Lightning, and mixed-precision training (using float32) is leveraged for efficiency.

3.5.4. Inference Process

Once trained, the model can be used to synthesize a novel view of an object from a given source image and a specified target camera pose.

Novel View Synthesis Pipeline:

The inference process, managed by the MVDPipeline, proceeds as follows:

1. **Inputs**: The pipeline takes a single source image, the desired target camera parameters, and an optional text prompt.

2. **Source Image Encoding**: The source image is processed through the ImageEncoder to extract its multi-scale visual features.

3. **Camera Parameter Encoding**: The target camera transformation is encoded through the CameraEncoder to produce a camera embedding.

4. **Iterative Denoising**: Starting from a randomly sampled Gaussian noise tensor in the latent space (of the same dimensions as the VAE's output latents), the MultiViewUNet iteratively denoises this latent over a predefined number of steps (in this case, 20). In each step, the U-Net receives the current noisy latent, the timestep t, the text prompt embedding, the extracted source image features, and the target camera embedding (via FiLM layers). Classifier-Free Guidance (CFG) is typically used, where the model makes both a conditional and an unconditional prediction, and the final noise estimate is a weighted combination, controlled by a guidance scale (set to 1.0).

5. **VAE Decoding**: After the final denoising step, the resulting clean latent representation is decoded back into pixel space using the frozen VAE's decoder.

3.5.5. Output

The final output of the pipeline is the synthesized image, which represents the object from the specified target viewpoint, conditioned by the appearance of the source image and guided by the geometric transformation. The output images are generated at a resolution consistent with the training data; the maximum resolution supported by the model is 768×768 pixels.

4. Data Preparation

In this chapter, I will discuss the datasets used in training the method presented in this thesis and the data augmentation techniques applied to them. The quality and nature of the data are paramount for training robust deep learning models, particularly for complex tasks such as novel view synthesis. This chapter details the meticulous process of curating and preparing the datasets used for training and validating the proposed method, with a primary focus on the extensive ObjaverseXL dataset used for training and the complementary Google Scanned Objects dataset used for validation.

4.1. Dataset Overview

Two primary datasets form the backbone of this work: ObjaverseXL, a large-scale repository of 3D models utilized for training the generative model, and Google Scanned Objects (GSO), a collection of high-fidelity 3D scans employed in my work for validation purposes.

4.1.1. ObjaverseXL: A Large-Scale 3D Model Repository

ObjaverseXL [5] is a vast, publicly accessible dataset comprising approximately 10 million 3D models. In order to use these models, one has to download them from their respective sources. The models are stored in diverse online sources, including platforms like GitHub, Thingiverse, and Sketchfab, as shown in 4.1, resulting in a rich and varied collection that reflects a wide array of object categories, complexities, and artistic styles. Given its scale and diversity, ObjaverseXL serves as the primary source for the training data in this thesis.

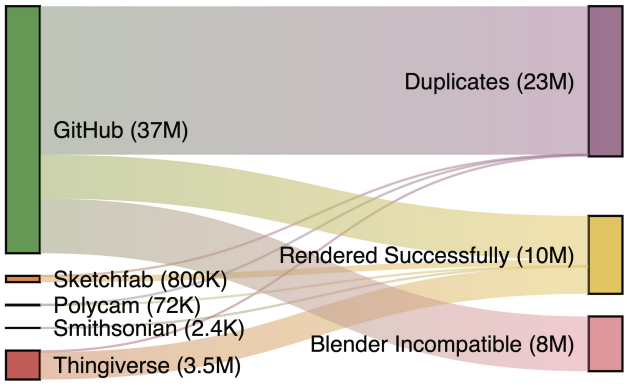

Figure 4.1: Illustrative overview of ObjaverseXL model sources and diversity.

4.1.2. Google Scanned Objects (GSO): High-Quality 3D Scans

The Google Scanned Objects (GSO) dataset [7] consists of high-fidelity 3D scans of real-world objects. In this work, the GSO dataset is utilized for the final validation of the trained multi-view image generation model. Its distinct origin and capture methodology compared to ObjaverseXL provide a benchmark for evaluating the generalization capabilities and output quality of the proposed method on unseen, high-quality data.

4.2. Processing Pipeline for ObjaverseXL Training Data

The transformation of raw 3D models from ObjaverseXL into a usable training dataset involved a comprehensive processing pipeline. This pipeline encompassed several critical stages: model acquisition, normalization, view rendering with specific lighting considerations, quality control filtering, and finally, textual annotation. Each step was carefully designed to address potential issues and ensure the final dataset's suitability for training an effective multi-view diffusion model.

4.2.1. 3D Model Acquisition and Initial Handling

The first step involved acquiring the 3D models from their respective sources as listed in the ObjaverseXL dataset. This task required scripting the download process due to the varied hosting platforms. The pipeline was designed to robustly handle common 3D model formats that typically support textures, including Object files (`.obj`), GLTF/GLB files (`.glb`, `.gltf`), Filmbox files (`.fbx`), and Collada files (`.dae`).

A critical part of initial handling was standardizing the coordinate system. 3D models from diverse origins often have different conventions for which axis represents "up". To ensure consistency, all imported mesh objects underwent a corrective rotation of -90 degrees around the X-axis. This transformation was immediately applied to the objects, establishing a common "Z-up" orientation for all models before subsequent processing steps. A preliminary inspection of model metadata was also performed upon successful download, which sometimes provided hints for orientation if the automated correction was insufficient, though the Z-up convention was paramount.

4.2.2. 3D Model Normalization and Scene Setup

To ensure consistency across all rendered images, a standardized scene setup was imperative. Each downloaded 3D model underwent a normalization process:

1. **Global Bounding Box Calculation**: The process began by calculating a single bounding box encompassing all mesh components of the loaded 3D model.

2. **Scaling to Unit Cube**: Based on the maximum dimension of this global bounding box, a uniform $scale_factor$ was determined. The model was then scaled by this factor to ensure it fit precisely within a 1x1x1 unit cube.

3. **Centering at Origin**: After scaling, the model was translated so that the center of its global bounding box was positioned at the world origin (0,0,0).

This normalization was critical for several reasons. Firstly, it ensured that objects, regardless of their original size, would fit within the camera's viewpoint. Secondly, it provided a consistent reference point for defining camera positions and distances, simplifying the multi-view rendering setup.

4.2.3. Multi-View Image Rendering

The core of the data generation process was rendering 2D images of each 3D model from multiple viewpoints.

Rendering Environment

All rendering tasks were performed using Blender [4], a powerful open-source 3D creation suite. To automate and scale the process, Blender was operated in windowless (headless) mode. The $BLENDER\_EEVEE$ rendering engine was chosen for its balance of speed and quality, making it suitable for generating a large volume of images. Render output was configured to $1024x1024$ pixels. Images were initially rendered with an alpha channel (e.g., to PNG format) to handle transparent backgrounds.

Specific EEVEE settings were configured to balance quality and performance while ensuring clear depiction of the objects:

- **Temporal Anti-Aliasing (TAA)**: Render samples set to 32 for smoother edges.
- **Ambient Occlusion (GTAO)**: Enabled to provide contact shadows and enhance perceived depth.
- **Bloom**: Disabled to prevent overly bright, glowing effects that could obscure object details.

Camera Viewpoint Configuration

Camera setup was designed for consistency and to provide a comprehensive view of the objects. Key camera parameters included:

- **Focal Length**: 35mm.
- **Sensor Width**: 32mm.
- **Camera Distance**: Cameras were positioned at a fixed distance of 1.8 Blender units from the world origin. Given the 1x1x1 object normalization, this distance ensured the object was well-framed.
- **Targeting**: A $TRACK\_TO$ constraint was applied to the camera, compelling it to always point towards the world origin (0,0,0), keeping the object centered.

To train a model capable of understanding objects from various perspectives and generalizing to different numbers of input views, images were rendered from a randomly selected set of predefined camera angles for each model. Three configurations were used for 6, 8 and 12 views, with specific azimuth (horizontal rotation) and elevation alternating between small positive and negative values to cover the object from all sides. The camera is focused on the origin. The placement is shown in 4.2 and 4.3.

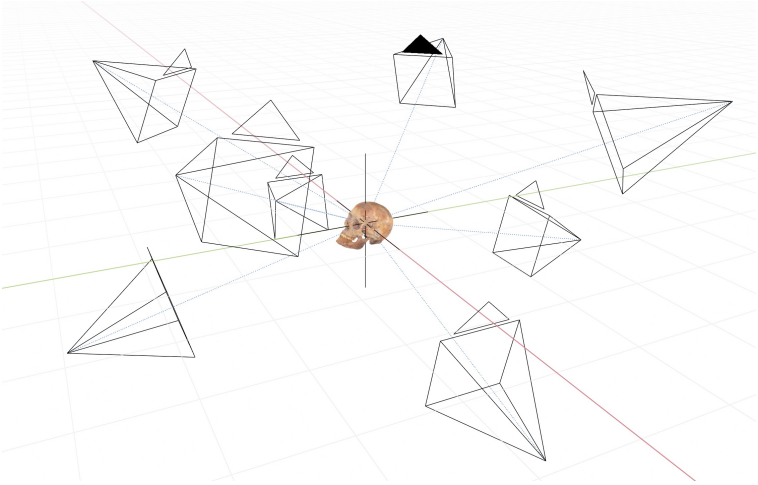

Figure 4.2: Example camera configurations for 8 views around an object.

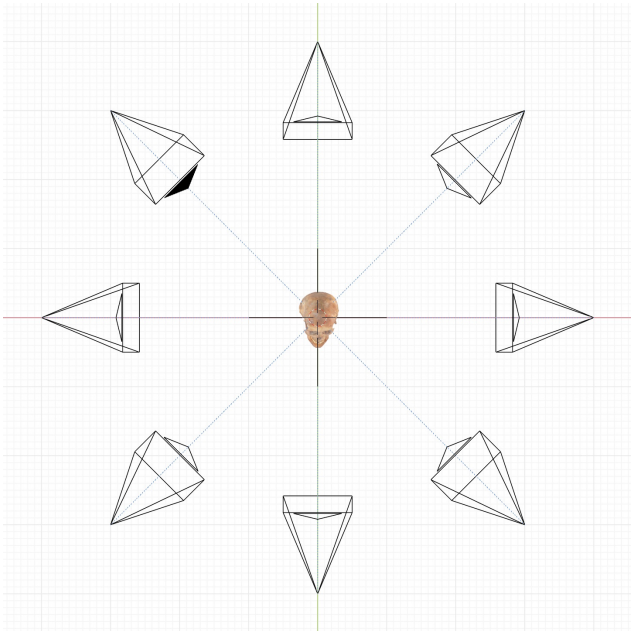

Figure 4.3: Camera configurations for 8 views around an object, visible from the top.

The choice of 6, 8, or 12 views for any given model was made randomly, with the dataset designed to have an approximately equal distribution of samples across these three groups. This strategy aimed to enhance the model's robustness to varying input view counts during

inference. Azimuth angles were defined to be counter-clockwise around the object to ensure compatibility with methods of 3D reconstruction like InstantMesh [34].

Lighting Strategy for Consistent Illumination

Lighting plays a critical role in the visual appearance of rendered objects. The example rendering scripts provided by the ObjaverseXL authors [6] employed a randomized lighting setup. While intended to introduce variability, this approach often resulted in images with harsh lighting, strong shadows, or scenes where the object was underexposed or even completely obscured because the light source was positioned unfavorably (e.g., behind the object relative to the camera). Such inconsistencies could negatively impact model training, potentially teaching the model to generate overly dark or inconsistently lit views.

To mitigate this, a revised, deterministic multi-point lighting strategy using four *SUN* type lights was adopted.The primary goal was to achieve soft, even illumination across the object's surface, ensuring that its features were clearly visible from all rendered viewpoints. This deterministic lighting strategy was specifically designed to create a dataset with consistent illumination, which helps address the 'Lighting issues' (Section 3.1 in Chapter 3) that can hinder the performance of multi-view generation models.

The resulting images featured more consistently well-lit objects, reducing the likelihood of the diffusion model learning to arbitrarily darken parts of an object from perspectives different from the main light source direction in the input. This consistency is vital for learning accurate geometry and appearance.

4.2.4. Post-Rendering Quality Control and Filtering

Despite the controlled rendering setup, not all rendering attempts yielded high-quality images. Some 3D models were inherently problematic (e.g., incompatible formats, missing textures, non-manifold geometry), while others, due to their material properties or fine details, resulted in renders that provided little useful visual information (e.g., images consisting mainly of reflections, transparency artifacts, or indiscernible silhouettes). It was observed that approximately one-third of the initially rendered samples suffered from such quality issues.

To ensure the training data was of sufficient quality, a rigorous filtering process was implemented. This was based on a contrast score, calculated as the standard deviation of pixel intensities in the grayscale version of each rendered image. A minimum contrast threshold was established. . If any of the rendered views for a particular 3D model fell below this threshold, the entire sample (the 3D model and all its associated renders) was discarded from the dataset. While this contrast metric doesn't capture all facets of 3D model quality, it proved to be a scalable and effective heuristic for discarding renders with insufficient visual information or severe artifacts. After this filtering stage, approximately $26,000$ high-quality 3D model samples, each with its set of 6, 8, or 12 views, remained.

4.2.5. Textual Prompt Generation via Multimodal LLM

To fine-tune a text-to-image diffusion model, rich textual descriptions (prompts) corresponding to the visual content are essential. Since the ObjaverseXL models do not inherently

come with such descriptive prompts, they needed to be generated. For this task, a state-of-the-art multimodal Large Language Model (LLM), Qwen2.5VL [1], was employed.

The prompt generation process was as follows:

1. For each 3D model, three views were randomly selected from its set of successfully rendered and filtered images.

2. These three views were provided as parallel input to the Qwen2.5VL model.

3. The LLM was tasked with analyzing these views and generating a single, comprehensive, and descriptive text prompt that accurately captured the essence of the 3D object depicted.

This approach aimed to distill visual information from multiple perspectives into a rich textual description, providing effective conditioning for the diffusion model during training. The use of multiple views was intended to help the LLM form a more holistic understanding of the object compared to relying on a single view.

4.3. Processing of Google Scanned Objects (GSO) Validation Data

The Google Scanned Objects (GSO) dataset was processed with a similar pipeline to ensure consistency in evaluation. The primary steps involved:

1. Downloading the GSO 3D models.

2. Passing them through the same normalization, multi-view rendering (using the same camera configurations and lighting strategy), quality control filtering pipeline and textual prompt generation described in Section 4.2.

4.4. Training Dataset Characteristics

Following the comprehensive processing pipeline, the final datasets were prepared for model training and validation. The ObjaverseXL training dataset consists of approximately 26,000 unique 3D models, each associated with a set of high-quality rendered views (6, 8, or 12 per model) and a descriptive textual prompt generated by Qwen2.5VL. This curated dataset provides a rich and diverse source of multi-view image-text pairs for training the diffusion model.

The GSO validation dataset comprises a set of consistently rendered views from the high-fidelity GSO models (approximately 1000 models), ready to be used for evaluating the performance of the trained model, particularly its ability to generalize to unseen objects and maintain multi-view consistency. With these meticulously prepared datasets, the subsequent stages of model training and experimentation, as detailed in the following chapters, could proceed on a solid foundation.

4.4.1. Dataset Statistics

This subsection presents a visual overview of key statistics derived from the processed ObjaverseXL dataset. These visualizations offer insights into the distribution of various data attributes, which are helpful in understanding the characteristics of the data used for training.

First, the distribution of the number of rendered views per 3D model is presented. As described in Section 4.2.3, models were rendered with 6, 8, or 12 views, with an approximately equal distribution. The exact frequencies (after filtering) are shown in Table 4.1.

Table 4.1: Distribution of Render Counts per Model.

Render Count	Frequency
6	9081
8	8261
12	8660

Figure 4.4 illustrates the distribution of file sizes (in Megabytes) of the original 3D models in the dataset.

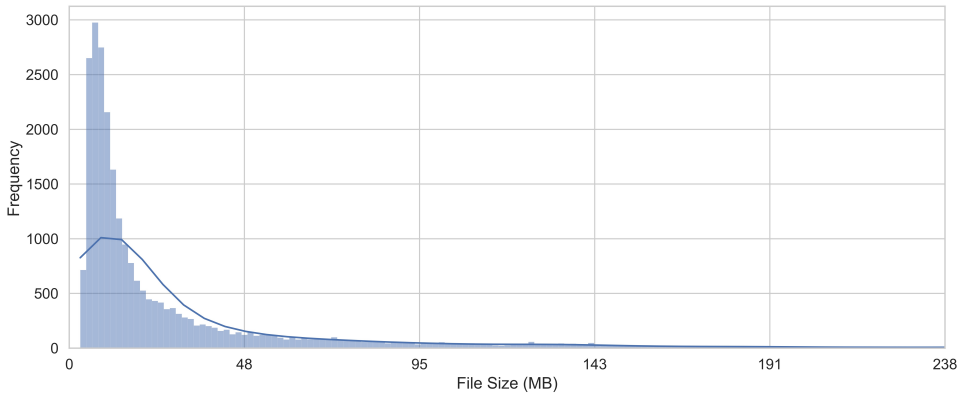

Figure 4.4: Distribution of sample sizes in the ObjaverseXL dataset. Statistics: $\mu = 37.25$ MB and $\sigma = 117.93$ MB.

The distribution of average image contrast across the rendered views is shown in Figure 4.5. This metric was crucial for the quality control filtering process, as detailed in Section 4.2.4.

Figure 4.6 displays the distribution of the lengths of textual prompts generated by the Qwen2.5VL model. These prompts are vital for conditioning the text-to-image diffusion model.

A word cloud visualization of the most frequent terms appearing in the generated prompts is presented in Figure 4.7. This offers a qualitative glimpse into the common themes and object characteristics described in the dataset.

Finally, Table 4.2 presents the top 20 topics identified from the textual prompts using Latent Dirichlet Allocation (LDA) [2]. Each topic is represented by its most characteristic words, providing insight into the semantic clusters present in the dataset's descriptions.

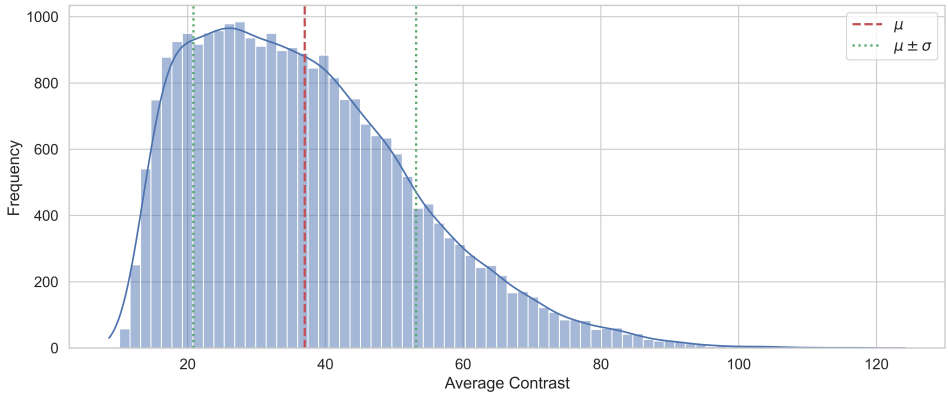

Figure 4.5: Distribution of average contrast scores for rendered images. Statistics: $\mu = 36.97$ and $\sigma = 53.15$.

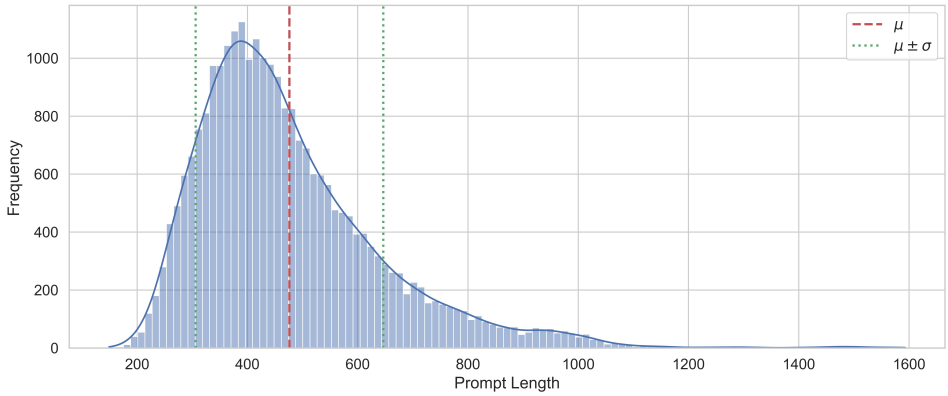

Figure 4.6: Distribution of generated prompt lengths. Statistics: $\mu = 476.06$ and $\sigma = 170.18$.

The identified topics highlight a diverse range of subjects, including common objects (e.g., 'box', 'chair', 'car'), structural elements (e.g., 'roof', 'legs', 'handle'), and descriptive attributes (e.g., colors like 'blue', 'red', 'green'; materials like 'metallic', 'wooden'; and characteristics like 'smooth', 'small'). This underscores the breadth of semantic content captured in the training data prompts.

These visualizations and statistics collectively provide a comprehensive overview of the dataset's characteristics, underscoring its suitability for the research presented in this thesis.

4.4.2. Dataset Samples

This section presents a selection of samples from the ObjaverseXL dataset. The samples are chosen to showcase the diversity of the dataset and the quality of the rendered views.

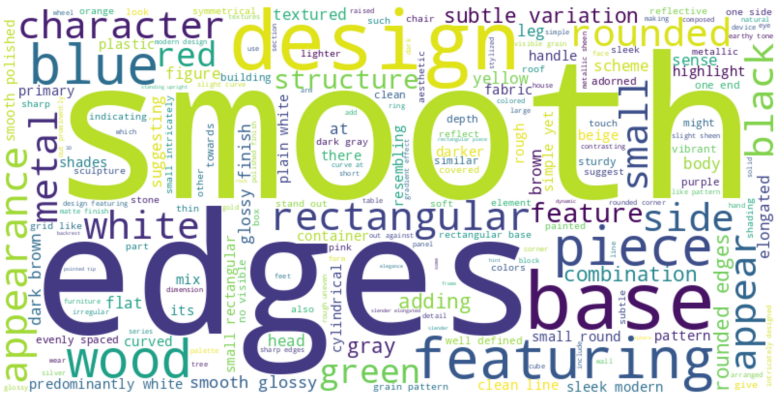

Figure 4.7: Word cloud of terms from generated prompts.

Table 4.2: Top 20 LDA Topics from Prompts with most characteristic words.

Topic	**Top Words**
#1	evenly, spaced, along, tall, easy, each, crate, rectangular, vertical
#2	device, rectangular, edges, rounded, black, small, industrial, panel
#3	character, small, body, figure, head, its, eyes, legs, black
#4	blue, red, glossy, smooth, vibrant, yellow, white, at, against
#5	body, sleek, streamlined, black, design, car, aerodynamic, accents, red
#6	metallic, metal, reflective, silver, polished, design, combination, gold, steel
#7	base, sculpture, intricately, ceramic, wear, statue, cylindrical, stone, signs
#8	box, ring, band, near, rectangular, black, pattern, either, sides
#9	green, block, small, tree, leaves, paper, delicate, plant, white
#10	well, defined, smooth, sharp, plain, skin, its, white, highlight
#11	roof, structure, small, rectangular, wooden, building, house, wood, painted
#12	legs, design, frame, chair, wood, backrest, sturdy, dark, seat
#13	rectangular, edges, modern, design, smooth, clean, base, white, lines
#14	smooth, subtle, brown, wood, rounded, edges, appearance, piece, variations
#15	section, part, upper, stick, base, lower, wider, than, more
#16	container, fabric, soft, rounded, lid, edges, small, pastel, suitable
#17	elongated, end, tapering, towards, rough, one, pointed, at, predominantly
#18	which, suggesting, might, indicating, similar, appears, plastic, could, like
#19	brown, shades, green, some, small, natural, rough, mix, areas
#20	handle, black, sleek, design, cylindrical, finish, modern, smooth, body

Figure 4.8: Sample image pairs from the ObjaverseXL dataset.

Figure 4.9: Sample views of Object 1 (6 views). Generated prompt: Create an image of a miniature lighthouse with a rustic vintage design. The base is made of dark wood with a polished finish, supporting a cylindrical tower made from a lighter material. The upper part features a lantern-like structure painted in a light color, adorned with small decorative elements like round windows or panels. The top is capped with a small, pointed roof made from the same light-colored material, slightly tilted.

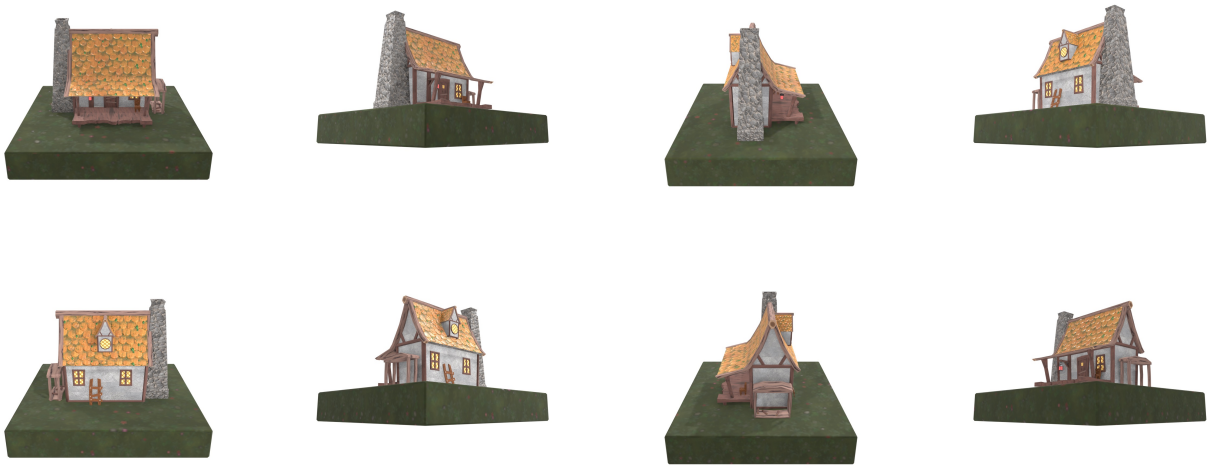

Figure 4.10: Sample views of Object 2 (8 views). Generated prompt: Create a detailed 3D model of a rustic cabin with a rectangular shape and slightly rounded corners. The cabin should have a warm, golden hue roof covered with shingles, wooden walls painted in a light color, small rectangular windows and door, and a raised foundation on stilts. The base should be a flat, green surface representing grass or another natural material. The overall aesthetic should be charming and rustic, with attention to detail in the textures and colors of the materials used.

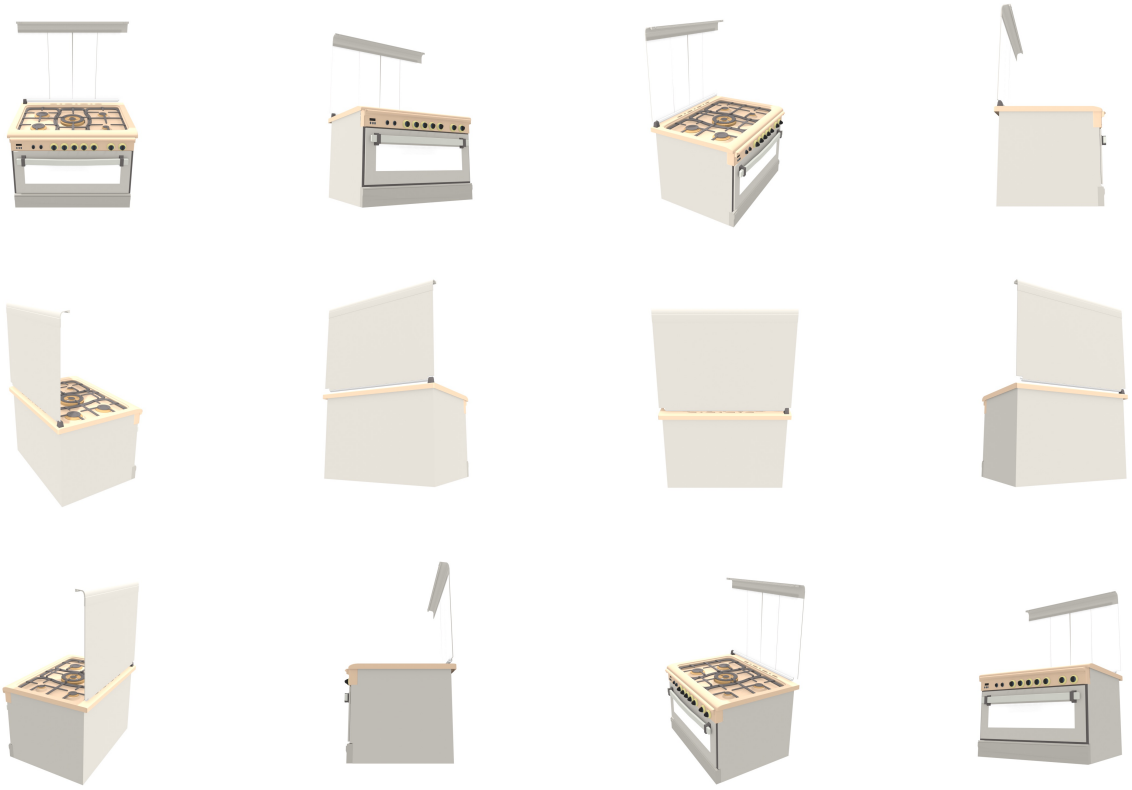

Figure 4.11: Sample views of Object 3 (12 views). Generated prompt: Create an image of a gas cooktop with an integrated oven. The design is sleek and contemporary. The cooktop section is made from a light-colored material, likely stainless steel, which reflects light and gives it a polished appearance. The hood extends outward, suggesting it is designed to cover the entire cooktop area. The edges of the hood are rounded, adding to the modern aesthetic. The color scheme is primarily metallic silver and black, with the black elements providing contrast against the silver surfaces.

5. Experiments

In this chapter, I describe the experimental validation of the proposed novel view synthesis method. The chapter begins by detailing the experimental setup, including data loading, evaluation metrics, and baseline training configurations. Subsequently, a series of targeted experiments are presented, designed to analyze the impact of different architectural components, conditioning strategies and training parameters. Each experiment's rationale, methodology, and results are discussed. The chapter also includes a qualitative analysis of generated images, a comparison with state-of-the-art methods where applicable, and an analysis of the model's inference efficiency.

5.1. Experimental Setup

This section outlines the common foundation for all experiments, including the data loading setup, the metrics for evaluation, and the standard training configuration of the proposed model.

5.1.1. Datasets

The experiments leverage two primary datasets, as detailed in Chapter 4:

- **Training Data**: The core training dataset is derived from ObjaverseXL [5], consisting of approximately 20,000 processed 3D models. For each model, a set of 6, 8, or 12 views were rendered at 1024×1024 resolution and subsequently resized to 768×768 for training. Textual prompts corresponding to these views were generated using the Qwen2.5VL multimodal LLM, as described in Section 4.2.5.
- **Validation and Test Data**: The Google Scanned Objects (GSO) dataset [7] serves as the testing set for evaluating generalization capabilities and for quantitative comparisons against state-of-the-art methods. Additionally, a small fraction of the processed ObjaverseXL dataset was reserved as an internal validation set to monitor training progress.

5.1.2. Data Loading and Preprocessing

The training and evaluation data, derived from the ObjaverseXL and GSO datasets, are processed and loaded as follows:

- **Dataset Splitting**: The collection of object files is deterministically split into training, validation, and test sets. This division is based on a specified ratio (e.g., 80% train, 10% validation, 10% test) applied after shuffling the list of all available object archives using

a fixed random seed. This ensures reproducibility of the splits across different runs and does not mix up the same object in different splits.

- **View Pair Generation**: For each object within a given split, pairs of source and target view pairs are constructed. They are randomly selected between each other, with a limit of 8 view pairs per object.

- **Individual Item Preprocessing**: When a specific view pair is constructed, the source and target images are loaded in a RGBA format, and then alpha-composited onto a white background before being converted to RGB. Images are resized to the specified $target_size$ (e.g., 768×768 pixels) using Lanczos resampling. The pixel values of the images are then normalized to the range [-1.0, 1.0] by scaling from [0, 255], since this is an expected format by VAE encoder.

This data loading pipeline ensures that the model receives consistently processed and structured input, suitable for training a Latent Diffusion Model.

5.1.3. Evaluation Metrics

To quantitatively test the performance of the novel view synthesis, the following metrics were employed:

- **Peak Signal-to-Noise Ratio (PSNR)**: Measures pixel-wise reconstruction accuracy. Higher is better.

- **Structural Similarity Index Measure (SSIM)** [32]: Assesses perceptual similarity based on luminance, contrast, and structure. Values range from -1 to 1, with 1 indicating perfect similarity.

- **Perceptual Loss**: Calculates the distance between feature activations in VGG16 model of generated and reference images, aligning better with human perception of image similarity [15]. Lower is better.

- **Learned Perceptual Image Patch Similarity (LPIPS)** [38]: Calculates the distance between deep features of generated and reference images, aligning better with human perception of image similarity. Lower is better.

- **Fréchet Inception Distance (FID)** [13, 29]: Compares the statistical similarity between distributions of generated images and real target views using InceptionV3 embeddings. Lower is better.

- **CLIP Score** [12]: Measures the cosine similarity between CLIP embeddings of the generated image and the target view. Higher is better.

5.1.4. Baseline Training Configuration

The proposed model, unless specified differently for a particular experiment, is trained using the following configuration:

- **Base Model Architecture**: Stable Diffusion 2.1, with the VAE and text encoder frozen. The U-Net is modified with the proposed image cross-attention adapters and FiLM layers for camera conditioning.
- **Trainable Parameters**: Only the newly introduced adapter layers, the $ImageEncoder$'s non-U-Net parts (if any beyond feature extraction), and the $CameraEncoder$ MLP and FiLM generator layers are trained. The original Stable Diffusion U-Net weights (when used for feature extraction in $ImageEncoder$ or as the backbone for the $MultiViewUNet$) remain frozen unless part of a full fine-tuning experiment.
- **Optimizer**: AdamW [20].
- **Learning Rate**: 1×10^{-5}.
- **Learning Rate Schedule**: Cosine annealing with a warm-up period of 500 steps.
- **Batch Size**: 6 per GPU, with gradient accumulation steps set to 1 (defined to change the effective batch size if needed).
- **Hardware**: 4 x NVIDIA A100 40GB GPUs.
- **Training Duration**: Typically 10 epochs for the 20k ObjaverseXL dataset, or a proportionally scaled number of steps for smaller datasets/experiments.
- **Image Resolution**: Input and output images are 768×768 pixels.
- **Default Conditioning Strengths**: img_ref_scale, $cam\_mod\_strength$.
- **Loss Function**: MSE loss with Min-SNR weighting strategy ($\gamma = 5.0$).
- **Classifier-Free Guidance (CFG) Scale (Inference)**: 1.0.
- **Inference Steps**: 20 steps using a DDPMScheduler.

5.2. Performed Experiments

This section presents a series of experiments designed to dissect the proposed architecture and study the impact of key design choices and training methodologies.

5.2.1. E1: Impact of Conditioning Strengths (img-ref-scale and cam-mod-strength)

Objective: To determine the optimal operating range and sensitivity of the model to the *img_ref_scale* (controlling influence of reference image features) and *cam_mod_strength* (controlling influence of camera FiLM modulation) hyperparameters.

Methodology: A hiperparameter tuning was performed by training separate models with various combinations of conditioning strength parameters. These experiments were conducted on a 5,000-sample subset of the ObjaverseXL dataset. The *img_ref_scale* values tested were {0.1, 0.25, 0.5, 0.75, 1.0} and *cam_mod_strength* values tested were {0.1, 0.25, 0.5, 0.75, 1.0}. Performance during these validation runs was evaluated using PSNR, SSIM, Perceptual Loss, FID, and CLIP score. The training loss was also monitored.

Results and Discussion: The impact of varying conditioning strengths on different evaluation metrics is presented below. The Fréchet Inception Distance (FID) is a key metric for assessing the quality of generated images. Figure 5.2 shows the effect of conditioning strengths on Perceptual Loss, SSIM, CLIP score, and FID.

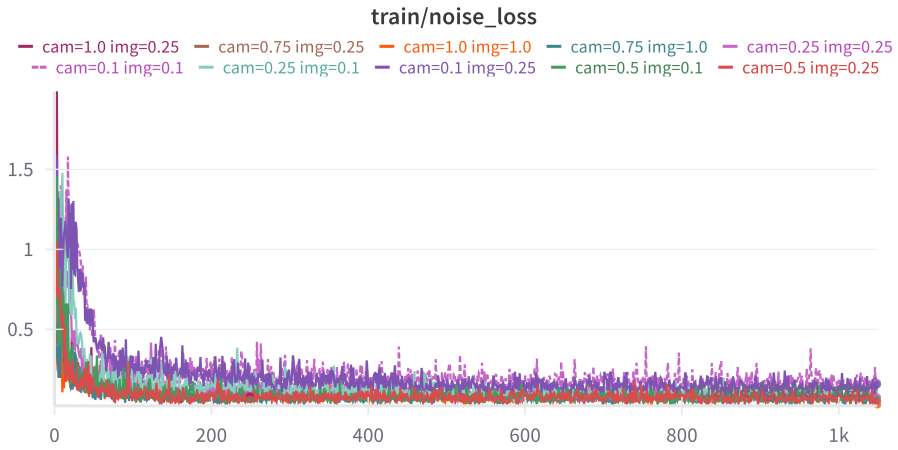

Figure 5.1: E1: Training MSE noise loss.

Impact of Camera Modulation Strength:

- Low Camera Strength: Consistently, low *cam_mod_strength* values, particularly 0.1, resulted in markedly inferior performance. These configurations exhibited the highest (worst) FID scores, generally above 170, and the lowest (worst) CLIP scores, around $0.68 - 0.69$, as well as the lowest SSIM values, around 0.88. This indicates that with weak camera conditioning, the model struggles to accurately interpret and apply the target view geometry, leading to poor synthesis quality and geometric inaccuracies.
- A *cam_mod_strength* of 0.25 showed an improvement over 0.1 but still significantly underperformed compared to stronger camera conditioning.
- High Camera Strength: Increasing *cam_mod_strength* to 0.5, 0.75, or 1.0 generally led to substantial improvements across FID, CLIP, and SSIM.

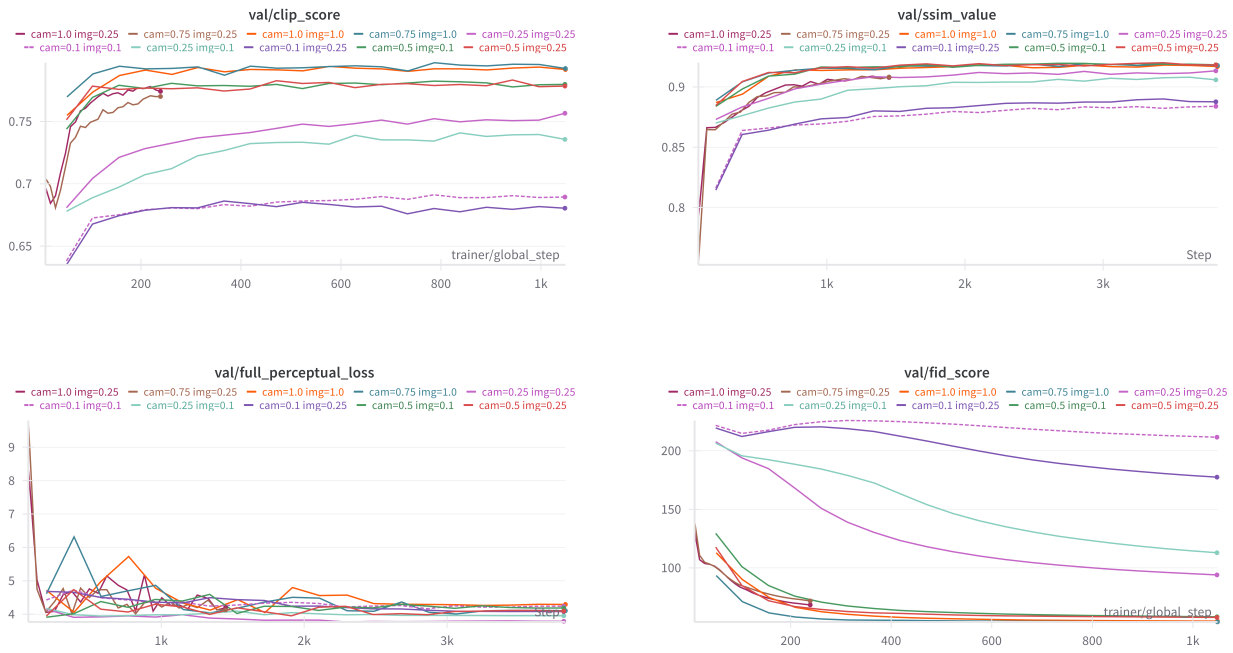

Figure 5.2: E1: Impact of conditioning strengths on various metrics: (a) CLIP Score, (b) SSIM, (c) Perceptual Loss, (d) FID Score.

Impact of Reference Image Conditioning Strength:

- Low Image Reference Strength: An img_ref_scale of 0.1, when paired with low to moderate $cam\_mod\_strength$, generally resulted in poorer FID, CLIP, and SSIM scores compared to higher img_ref_scale values. This suggests that some degree of visual information from the source view is beneficial for guiding the synthesis process.

- Moderate Image Reference Strength: An img_ref_scale of 0.25 emerged as a particularly effective value, especially for optimizing FID. When combined with adequate $cam\_mod\_strength$ (0.5 or higher), configurations such as $cam = 0.75 img = 0.25$, $cam = 1.0 img = 0.25$ and $cam = 0.5 img = 0.25$ consistently achieved among the lowest FID scores (around 50-60). These combinations also yielded strong CLIP scores (approx. 0.76-0.78) and SSIM values (approx. 0.91-0.925).

- High Image Reference Strength: An img_ref_scale of 1.0, when paired with high $cam\_mod\_strength$, also demonstrated top-tier performance, particularly excelling in CLIP score (reaching up to ≈ 0.79) and SSIM (reaching up to ≈ 0.93). However, these configurations sometimes resulted in slightly higher (worse) FID scores compared to their $img\_ref\_scale = 0.25$ counterparts with similar high $cam\_mod\_strength$. This might suggest that while a strong reference image signal can enhance perceptual similarity and structural details (benefiting CLIP and SSIM), it might slightly constrain the model's ability to generate the most statistically faithful novel view if it adheres too closely to the source image features.

This hyperparameter sweep demonstrates that the model is sensitive to both $cam\_mod\_strength$ and img_ref_scale. For the most robust novel view synthesis, a $cam\_mod\_strength$ of at least 0.5 is recommended, with 0.75 or 1.0 often yielding the best results. For img_ref_scale, a value of 0.25 provides an excellent balance, particularly for FID, while $img\_ref\_scale = 1.0$ can also be highly effective, especially for maximizing CLIP and SSIM scores when paired with strong camera conditioning. In the future experiments, the parameters were set to $img\_ref\_scale = 0.75$ and $cam\_mod\_strength = 1.0$.

5.2.2. E2: Training Strategy: Adapters vs. Full Fine-tuning

Objective: To compare the performance, training efficiency (time and computational resources), and parameter efficiency of the proposed adapter-based training approach against full fine-tuning of the U-Net. This addresses the "Training adapter only vs full fine-tuning" point.

Methodology: Two main training strategies were compared. The first approach involved training only the lightweight adapter modules and camera conditioning encoder (adapter only training), keeping the backbone Stable Diffusion U-Net frozen. For a direct comparison on a smaller scale, this was performed on a 5,000-sample subset of ObjaverseXL for 10 epochs.

The second approach involved fine-tuning the entire U-Net along with the adapters and conditioning encoders (full U-Net fine-tuning). This was performed on a 5,000-sample subset of ObjaverseXL for 10 epochs for comparison with the adapter-only training on the same dataset size.

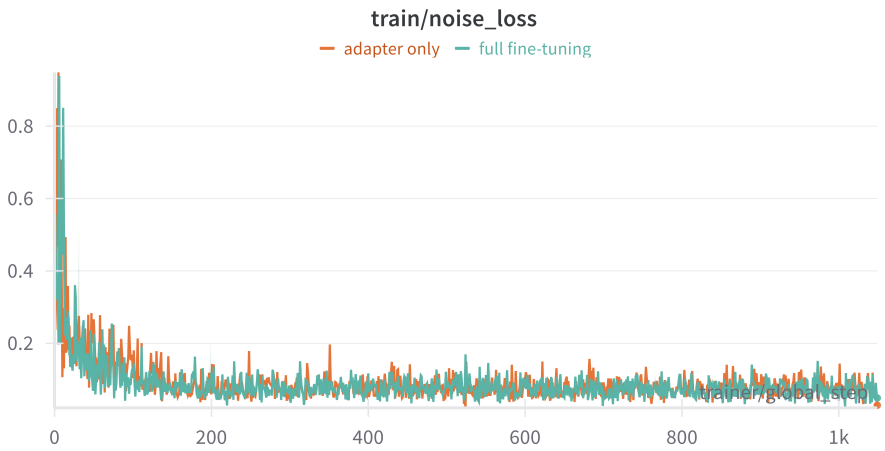

Figure 5.3: E2: Training MSE noise loss.

Results and Discussion: Training the adapter only was possible with batch size of 6 (per GPU) and full fine-tuning was possible with batch size of 2 (per GPU) with gradient accumulation steps set to 3 to match the effective batch size between the two approaches.

Out of the $2.9B$ parameters of the full model (U-Net + VAE + Text Encoder + Adapters), $585M$ are trainable in the adapter only method and $2.3B$ are trainable in the full fine-tuning method.

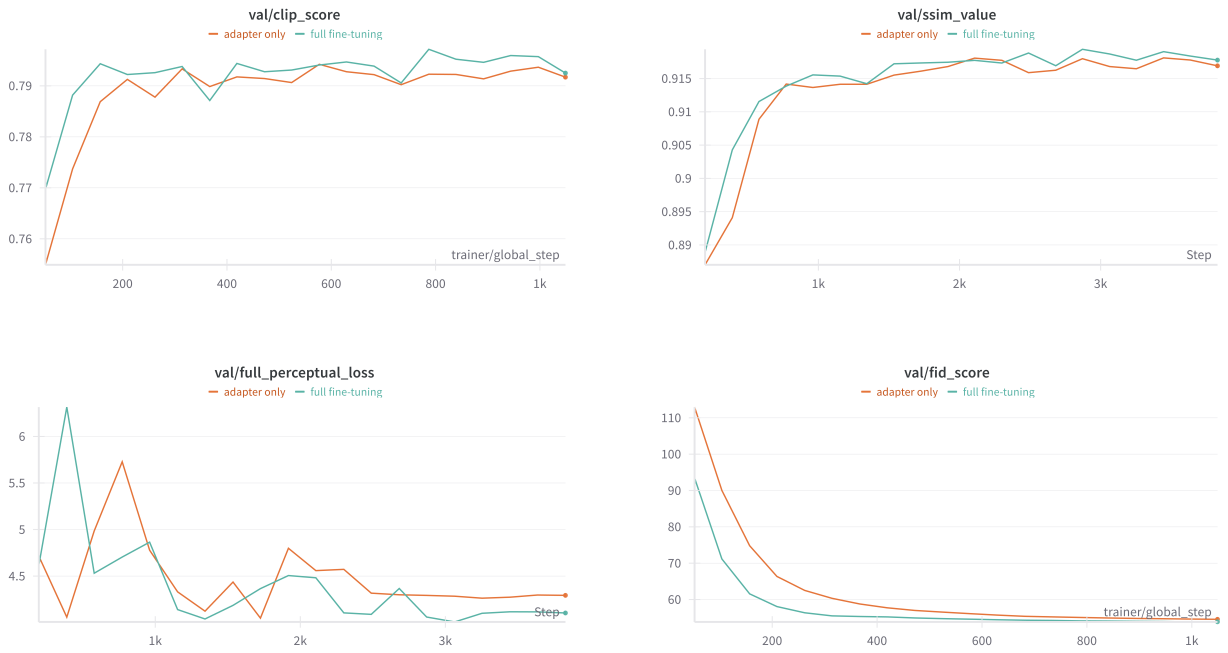

Figure 5.4: E2: Training adapters only vs. full fine-tuning of U-Net on various metrics: (a) CLIP Score, (b) SSIM, (c) Perceptual Loss, (d) FID Score.

The adapter only method, trained for 10 epochs took 9 hours on 4 x NVIDIA A100 40GB GPUs. The full fine-tuning method, trained for 10 epochs took 35 hours on the same hardware. This is a significant difference in training time, but the adapter only method is still able to achieve a comparable performance to the full fine-tuning method.

The quantitative results on auxiliary metrics are shown in Figure 5.4. The full fine-tuning method is able to achieve a slightly better performance on the Perceptual Loss metric, but the adapter only method achieves a comparable performance based on the SSIM, CLIP Score, and FID metrics.

5.2.3. E3: Camera Encoder Architecture Details

Objective: To investigate the impact of the *CameraEncoder*'s architectural design on novel view synthesis performance. Specifically, this experiment explores two main factors: (1) the depth of the Multi-Layer Perceptrons (MLPs) used to process relative rotation and translation camera parameters, and (2) the dimensionality of the internal hidden representations and the final camera embedding used for FiLM modulation.

Methodology: The *CameraEncoder* module computes a relative transformation (rotation matrix R and translation vector T) between source and target camera poses. The translation vector undergoes positional encoding. Both R (flattened) and the positionally encoded T are then processed by separate MLP streams before their outputs are concatenated and passed through a final projection MLP to produce the camera embedding. Four configurations of the *CameraEncoder* were evaluated, varying MLP depth and embedding dimensionality:

- **Shallower MLPs, Lower Dimensionality**: Rotation and translation encoder MLPs each consist of two linear transformation stages. The internal hidden dimension within these MLPs is 512, and the final camera embedding dimension is 1024.
- **Shallower MLPs, Higher Dimensionality**: Rotation and translation encoder MLPs each consist of two linear transformation stages. The internal hidden dimension is 1024, and the final camera embedding dimension is 2048.
- **Deeper MLPs, Lower Dimensionality**: Rotation and translation encoder MLPs each consist of three linear transformation stages. The internal hidden dimension is 512, and the final camera embedding dimension is 1024.
- **Deeper MLPs, Higher Dimensionality**: Rotation and translation encoder MLPs each consist of three linear transformation stages. The internal hidden dimension is 1024, and the final camera embedding dimension is 2048.

All four model variants, differing only in their *CameraEncoder* configuration as described, were trained using the adapter-only approach on a 5,000-sample subset of the ObjaverseXL dataset for 10 epochs.

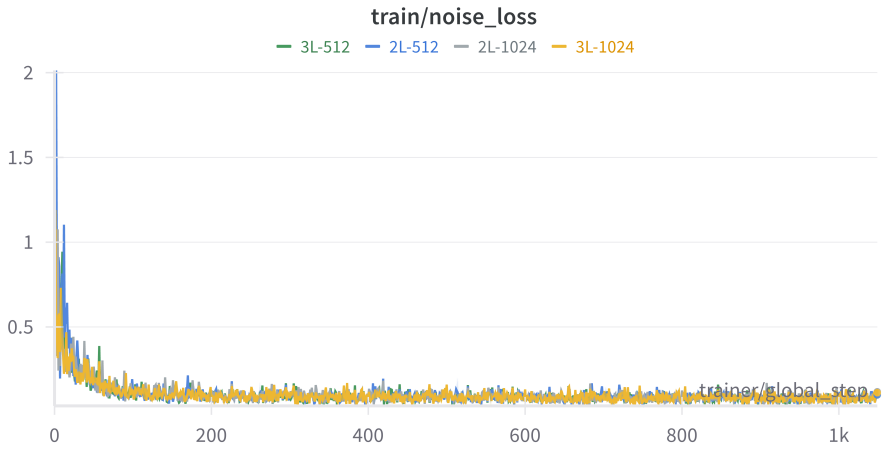

Figure 5.5: E3: Training MSE noise loss.

Results and Discussion: The quantitative results demonstrate consistent performance across all four */textCameraEncoder* architectural variants, as shown in Figure 5.6. All configurations converge to similar final values: CLIP scores of $\approx 0.78 - 0.79$, SSIM values of $\approx 0.92 - 0.93$, FID scores of $\approx 103 - 106$, and perceptual loss of $\approx 3.6 - 3.7$. This consistency suggests that architectural complexity has minimal impact on camera parameter encoding effectiveness.

Detailed analysis reveals:

- **MLP Depth**: Deeper 3-layer configurations (3L-512, 3L-1024) show no substantial advantages over 2-layer counterparts (2L-512, 2L-1024). Performance differences remain within typical training variance, indicating that additional depth does not enhance camera transformation encoding.

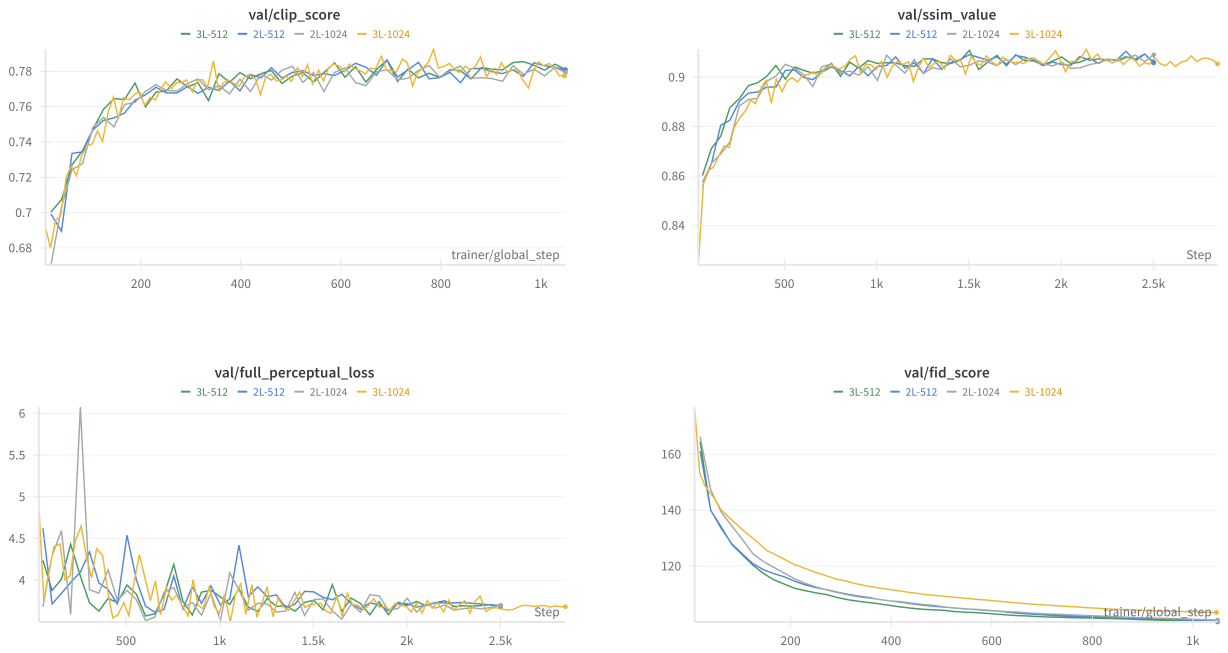

Figure 5.6: E3: Impact of camera encoder depth on various metrics: (a) CLIP Score, (b) SSIM, (c) Perceptual Loss, (d) FID Score.

- **Embedding Dimensionality**: Increasing hidden dimensions from 512 to 1024 and final embeddings from 1024 to 2048 produces negligible performance improvements. Lower-dimensional representations appear sufficient for capturing necessary geometric relationships.

- **Training Dynamics**: All variants exhibit similar convergence patterns and stable validation metrics, demonstrating that FiLM-based conditioning is robust to moderate architectural variations.

- **Marginal Differences**: 3L-512 achieves slightly better FID performance.

These findings indicate that the camera encoding bottleneck lies in the fundamental challenge of mapping geometric transformations to effective feature modulations, rather than architectural complexity. The baseline 2L-512 configuration provides optimal efficiency while maintaining competitive performance, validating its use in subsequent experiments.

5.2.4. E4: Impact of Training Data Scale

Objective: To investigate how the size of the training dataset (number of unique 3D objects from ObjaverseXL) influences the modelś performance, particularly its generalization ability to unseen objects (GSO dataset) and overall robustness. This is especially relevant for addressing the ”Adaptation to unseen objects” limitation of current state-of-the-art methods.

Methodology: The proposed model, using the adapter-only training approach and consistent conditioning strengths where possible, was evaluated on increasingly larger subsets of the processed ObjaverseXL dataset:

- **1,000 samples** for 10 epochs.
- **5,000 samples** for 10 epochs.
- **20,000 samples** for 10 epochs.

All models were trained for a comparable number of effective updates relative to their dataset size where possible (e.g., by adjusting epoch counts or comparing at similar total iterations/epochs across all scales). Performance was evaluated on the GSO test set using all key metrics.

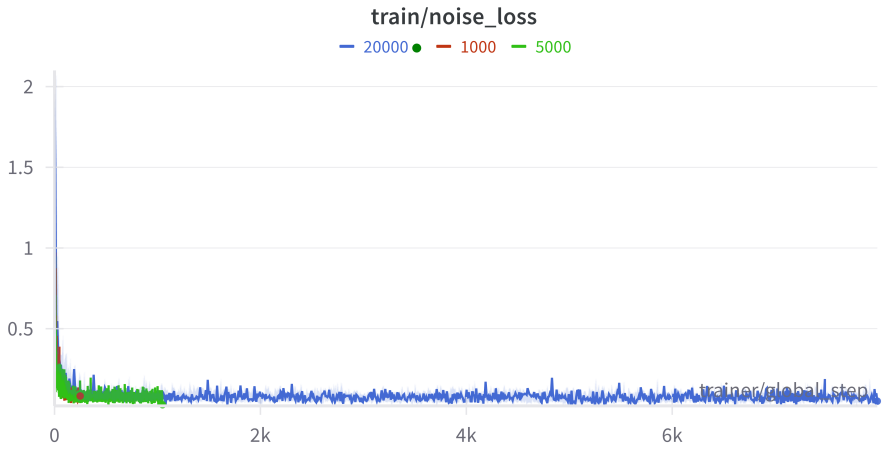

Figure 5.7: E4: Training MSE noise loss.

Results and Discussion: The results demonstrate a clear positive correlation between training data scale and model performance across all evaluation metrics, as shown in Figure 5.8.

Performance Scaling with Data Size:

- **1,000 Samples**: The smallest dataset configuration achieved the lowest performance across all metrics. CLIP scores plateaued around 0.77, SSIM values reached approximately 0.91, FID scores remained elevated at around 70, and perceptual loss fluctuated around 4.5. This indicates that insufficient training data limits the model's ability to learn robust visual-geometric relationships necessary for high-quality novel view synthesis.
- **5,000 Samples**: Scaling up to 5,000 samples showed notable improvements across all metrics. CLIP scores improved to approximately $0.78 - 0.79$, SSIM values increased to around 0.92, FID scores decreased to approximately 60, and perceptual loss reduced to around 4.2. This demonstrates that increased data diversity enhances the model's understanding of 3D object appearance and geometry.

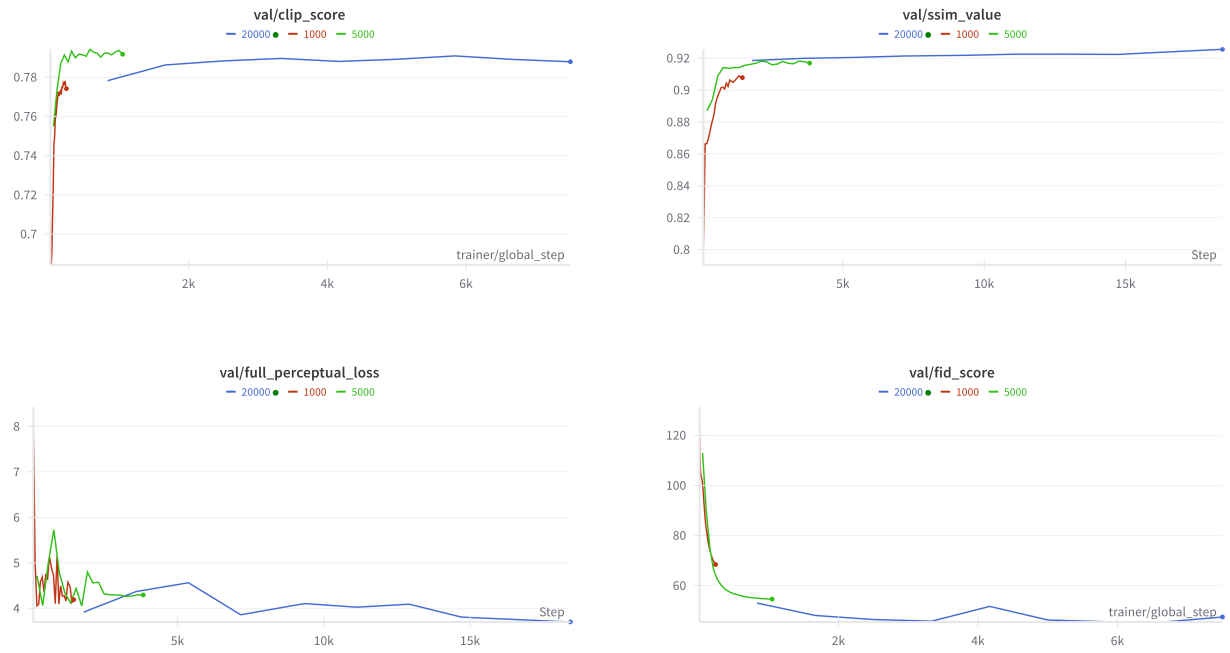

Figure 5.8: E4: Impact of training data scale on various metrics: (a) CLIP Score, (b) SSIM, (c) Perceptual Loss, (d) FID Score.

- **20,000 Samples**: The largest dataset configuration achieved the best performance across all metrics. CLIP scores reached the highest values of approximately 0.79, SSIM peaked at around 0.93, FID achieved the lowest scores of approximately 45, and perceptual loss reached the minimum values of around 3.7. This configuration also demonstrated the most stable training dynamics, as evidenced by the smoother convergence curves.

Training Dynamics and Convergence:

- **Convergence Speed**: Larger datasets exhibited faster and more stable convergence patterns. The 20,000-sample configuration showed the smoothest training loss curves with minimal fluctuations, while smaller datasets displayed more erratic training dynamics.
- **Generalization**: The performance improvements observed with larger training sets directly translated to better generalization on the GSO test set, indicating that increased data diversity helps the model adapt to unseen object geometries and appearances.
- **Training Stability**: Models trained on larger datasets showed reduced variance in validation metrics across different training runs, suggesting improved robustness to initialization and sampling variations.

Data Scaling Implications: The consistent improvement across all metrics with increased training data size validates several key hypotheses:

- **Diversity Benefits**: Larger datasets provide greater diversity in object shapes, textures, and lighting conditions, enabling the model to learn more generalizable representations for novel view synthesis.

- **Geometric Understanding**: The improvement in SSIM and perceptual metrics indicates that more training data enhances the model's ability to preserve structural details and geometric consistency across viewpoints.
- **Distribution Matching**: The reduction in FID scores with larger datasets suggests that the model better captures the statistical properties of real image distributions when provided with more diverse training examples.

These findings strongly support the importance of large-scale training data for robust diffusion-based novel view synthesis and suggest that further scaling beyond 20,000 samples could yield additional performance improvements. However, even the largest configuration tested (20,000 samples) remains substantially smaller than state-of-the-art methods like Zero123++, which utilized $800K+$ 3D models - indicating that the observed performance limitations may be fundamentally constrained by data scale rather than architectural choices. The 20,000-sample configuration was selected as the baseline for subsequent experiments and final model evaluation within the available computational constraints.

5.3. Comparison with State-of-the-Art Methods on GSO Dataset

This section presents the quantitative performance of the final proposed model on the Google Scanned Objects (GSO) dataset and compares it against relevant state-of-the-art methods.

5.3.1. Performance of the Proposed Method

To comprehensively evaluate the proposed novel view synthesis method, three model configurations representing different training data scales were assessed on the GSO dataset. These configurations were selected to demonstrate the impact of training data size on generalization performance:

- **Configuration 1**: Trained on 1,000 ObjaverseXL samples for 10 epochs.
- **Configuration 2**: Trained on 5,000 ObjaverseXL samples for 10 epochs.
- **Configuration 3**: Trained on 20,000 ObjaverseXL samples for 10 epochs.

All models were trained using the adapter-based approach with optimal hyperparameters ($img\_ref\_scale = 1.0$ and $cam\_mod\_strength = 1.0$) identified in previous experiments.

The evaluation protocol followed a standardized procedure where each model received a source view and the corresponding camera transformation parameters to synthesize a target novel view. The generated images were then quantitatively compared against ground truth target views using the established evaluation metrics. All assessments were conducted on a held-out test set comprising 100 carefully selected samples from the GSO dataset, ensuring no overlap with training data and providing a robust benchmark for cross-dataset generalization capabilities.

5.3.2. Comparison with State-of-the-Art Methods

The performance of the proposed method is compared against two contemporary novel view synthesis techniques: Zero123++ and MVAdapter. The results, evaluated on 100 samples from the GSO dataset with aligned camera angles and consistent conditioning (reference image and text prompt), are presented in Table 5.1. The comparison reveals significant performance gaps, with the proposed method underperforming established approaches across key metrics. Notably, these performance differences must be contextualized by the substantial disparity in training data scale: Zero123++ was trained on 800K+ Objaverse models with approximately 10M rendered images, while the proposed method used only 20,000 models - representing a 40× difference in 3D model diversity and likely 100× difference in training images. This fundamental difference in data scale significantly impacts generalization capabilities to unseen objects.

Table 5.1: Quantitative comparison with state-of-the-art methods on 100 samples from the GSO dataset.

Method	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
Zero123++ [19]	**19.27**	**0.84**	**0.12**
MVAdapter [14]	11.48	0.78	0.16
Proposed Method (1k samples, 10 epochs)	10.12	0.76	0.24
Proposed Method (5k samples, 10 epochs)	13.12	0.81	0.20
Proposed Method (20k samples, 10 epochs)	*13.48*	*0.82*	*0.18*

5.3.3. Inference Time Analysis

Objective: To quantify the computational efficiency of the proposed model during the inference phase (novel view generation).

Methodology: The average inference time was measured for generating a single novel view at the target resolution of 768 × 768 pixels.

- **Hardware Used**: Single NVIDIA A100 40GB GPU.
- **Model Configuration**: The best performing adapter-based model (trained on 20,000 ObjaverseXL samples).
- **Inference Steps**: 20 diffusion steps.
- **Batch Size for Inference**: 1 (generating one view at a time).

The results will be compared to the inference time of other state-of-the-art methods Zero123++ [19] and MVAdapter [14] (with their default settings).

Results and Discussion: The proposed adapter-based model achieves an average inference time of 16.04 seconds per 768 × 768 image on an NVIDIA A100 GPU, using 20 denoising steps. This suggests that the proposed model is able to generate novel views in comparable time to the MVAdapter method. The Zero123++ method is still significantly faster, with an average inference time of 6.51 seconds.

Table 5.2: Inference time comparison with state-of-the-art methods.

Method	**Inference Time (seconds)** ↓
Zero123++ [19]	6.51
MVAdapter [14]	19.42
Proposed Method	16.04

5.4. Qualitative Results

This section presents visual examples of novel view synthesis results generated by the proposed method on the GSO test dataset. Each comparison image shows three views arranged side by side: the source (reference) image on the left, the ground truth target view in the middle, and the synthesized novel view generated by the proposed method on the right. The selected samples demonstrate the method's performance across various object categories, geometric complexities, and viewpoint transformations present in the GSO dataset.

Figure 5.9: Sample 1: The generated human figure maintains the basic pose and structure but lacks fine anatomical details.

Figure 5.10: Sample 2: The spacecraft model shows reasonable overall shape preservation but exhibits color shifts and reduced detail fidelity in the generated view.

Figure 5.11: Sample 3: The chair synthesis demonstrates good structural understanding but suffers from texture and structural degradation.

Figure 5.12: Sample 4: The satellite 3D reconstruction model generation captures the basic geometric layout but produces notable artifacts in textural information and color.

Figure 5.13: Sample 5: The humanoid robot figure maintains correct proportions and pose but shows reduced detail sharpness and some color inconsistencies.

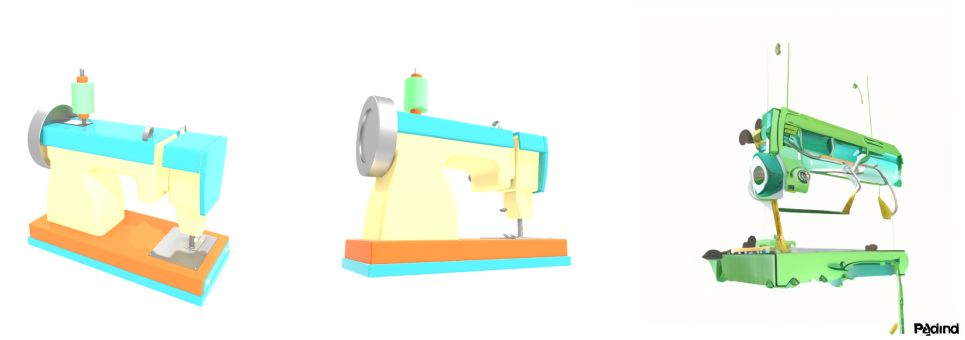

Figure 5.14: Sample 6: The sewing machine synthesis successfully preserves the overall structure and major components but lacks the crisp detail definition of the target view and includes a logo on the bottom right side.

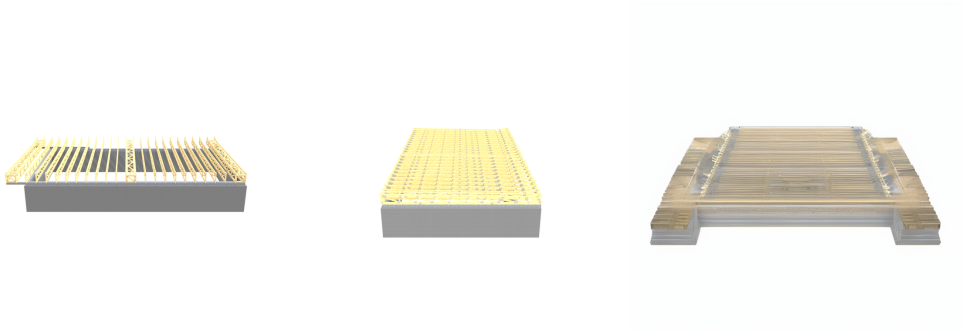

Figure 5.15: Sample 7: The wooden deck structure shows acceptable geometric consistency but exhibits noticeable texture blurring and reduced material realism.

Figure 5.16: Sample 8: The construction framework scene is too complex for the model, the generated view shows significant detail loss and simplified material representation.

Figure 5.17: Sample 9: The pineapple synthesis captures the basic shape and positioning but exhibits reduced color vibrancy and simplified surface texture patterns.

Figure 5.18: Sample 10: The house structure shows reasonable architectural preservation but displays noticeable loss of fine architectural details and color.

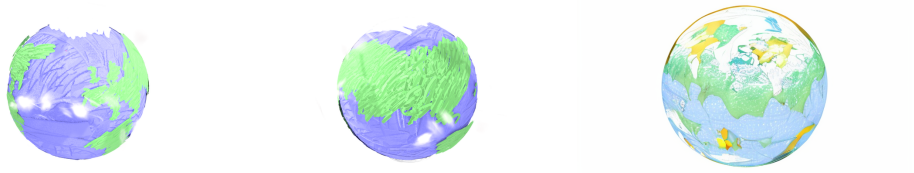

Figure 5.19: Sample 11: The Earth globe generation maintains spherical shape and continental outlines but shows reduced color saturation and simplified surface features.

The qualitative results reveal several consistent patterns in the model's performance limitations. While the proposed method demonstrates a fundamental understanding of 3D geometry and can generate plausible novel views that preserve basic object structure and pose, the generated images consistently suffer from reduced visual fidelity compared to ground truth targets. Common issues include: (1) texture degradation and loss of fine surface details, (2) color desaturation and shifts that reduce material realism, (3) overall blurriness that diminishes image sharpness, and (4) simplified representations of complex geometric features.

The model shows particular strength in maintaining overall object proportions and spatial relationships, indicating successful camera conditioning, but struggles with high-frequency details and photorealistic texture synthesis. These visual limitations align closely with the quantitative metrics, where lower PSNR and SSIM scores reflect the reduced pixel-level accuracy and structural preservation, while higher perceptual loss values indicate the model's difficulty in capturing perceptually important visual features that distinguish high-quality novel view synthesis.

5.5. Chapter Summary

This chapter detailed the comprehensive experimental evaluation of the proposed multi-view novel view synthesis method. The experimental setup, including datasets (ObjaverseXL for training, GSO for testing) and evaluation metrics (PSNR, SSIM, LPIPS, FID, CLIP score), was established.

Key findings from the experiments include:

- **Optimal Conditioning Strengths**: The investigation into img_ref_scale and $cam\_mod\_strength$ (Section 5.2.1) identified optimal ranges for these hyperparameters, demonstrating a balance is needed to effectively incorporate conditioning signals without introducing artifacts. The combination of $img\_ref\_scale = 1.0$ and $cam\_mod\_strength = 1.0$ was found to be effective.

- **Efficiency of Adapter-Based Training**: The comparison between adapter-only training and full U-Net fine-tuning (Section 5.2.2) highlighted the significant advantages of the adapter approach in terms of parameter efficiency and training resource reduction, while achieving competitive performance.

- **Camera Encoder Design**: Experiments with the *CameraEncoder* architecture (Section 5.2.3) suggested that the baseline moderately complex encoder offered a good balance of performance and efficiency.

- **Impact of Data Scale**: Increasing the training data size (Section 5.2.4) generally led to improved performance and generalization on the GSO dataset, underscoring the importance of large-scale diverse data for training robust NVS models.

Quantitative evaluation on the GSO dataset (Section 5.3) revealed substantial performance gaps compared to SOTA approaches like Zero123++ and MVAdapter. The proposed method achieves PSNR scores approximately 30% lower than Zero123++, indicating significant room for improvement in reconstruction quality. The FID score of 45, while improved from smaller

training datasets, remains considerably high for image generation tasks and suggests that further training is needed to improve the quality of the generated images.

While the experiments provide insights into architectural design choices and training strategies, the results highlight the challenges in achieving competitive performance for diffusion-based novel view synthesis within resource constraints. The significant underperformance compared to established methods can be largely attributed to the substantial difference in training data scale - Zero123++ leveraged 40× more 3D models and approximately 100× more training images than the proposed method. This disparity fundamentally limits the model's ability to generalize across diverse object types and geometric configurations, as evidenced by the performance gaps on the GSO dataset. Future work must address these limitations either through access to larger-scale training data, more sophisticated architectures that can better leverage limited data, or hybrid approaches that combine the demonstrated efficiency benefits with improved synthesis capabilities.

6. Conclusions

This thesis presents a novel approach to diffusion-based novel view synthesis combining FiLM-based camera conditioning with adapter-based training. This chapter summarizes the key findings, discusses limitations, and outlines future research directions.

6.1. Summary of Findings

1. **Hybrid conditioning strategy** combining visual and geometric information shows internal consistency improvements, with optimal configuration ($img\_ref\_scale = 1.0$, $cam\_mod\_strength = 1.0$) demonstrating complementary information fusion, though overall performance significantly lags behind established methods like Zero123++.

2. **Adapter-based training** achieves computational efficiency gains, training only $585M$ parameters (20% of full model) with $4\times$ faster training time, but this efficiency comes at the cost of synthesis quality, as evidenced by substantial performance gaps in quantitative evaluation.

3. **FiLM-based camera conditioning** provides a computationally efficient alternative to complex raymap representations, though the achieved FID scores of 45-50 remain substantially higher than state-of-the-art methods, indicating limited synthesis quality despite artifact reduction compared to raymap approaches.

4. **Dual-stream conditioning mechanism** demonstrates architectural feasibility for processing reference images through parallel conditioning streams, though the resulting novel views show significant quality limitations when compared to contemporary approaches.

6.2. Limitations and Areas for Improvement

The experimental results reveal fundamental performance limitations that must be acknowledged:

Resolution Constraints: The current implementation operates at a maximum resolution of 768×768 pixels, which may be insufficient for applications requiring high-detail novel view synthesis.

Single-Object Focus: The method is primarily designed and evaluated for single-object novel view synthesis. Extension to complex scenes would require architectural modifications and different training strategies.

Limited Viewpoint Range: While the training data covers comprehensive viewpoint distributions 6, 8 or 12 views, extreme viewpoint changes (such as complete 180-degree

rotations or significant elevation changes) may still pose challenges for geometric consistency, particularly for objects with complex occlusions or view-dependent materials.

Constrained Training Scale and Generalization: The most significant limitation is the substantial disparity in training data compared to state-of-the-art methods. While the proposed method used 20,000 Objaverse samples, Zero123++ was trained on 800K+ Objaverse models with approximately 10M rendered images - representing a 40× difference in 3D model diversity and 100× difference in training scale. This fundamental data limitation constrains the model's ability to generalize across diverse object types not well-represented in the limited training set, explaining much of the observed performance gap.

Inference Speed: While competitive with similar methods, the 16-second inference time may be prohibitive for real-time applications or interactive systems requiring immediate novel view generation.

6.3. Future Research Directions

Several promising directions emerge from recent research:

- **Advanced architectural foundations**: Adapting the proposed FiLM-based conditioning to modern diffusion transformer architectures (Stable Diffusion 3 [8, 22]) could potentially improve synthesis quality through more powerful attention mechanisms, text enrichment, flow matching and scaling properties.
- **Integrated 3D reconstruction pipelines**: Developing hybrid systems that combine explicit 3D reconstruction with generative synthesis could leverage both geometric and generative methods, reconstructing coarse 3D representations then using diffusion models to generate high-quality details such as Human3Diffusion [35].
- **Multi-view consistency and real-time optimization**: Future work could explore stronger geometric constraints for coherent multi-view synthesis and investigate model distillation or quantization for real-time applications in VR/AR and interactive systems.

The work presented in this thesis explores the potential of efficient adapter-based approaches for diffusion-based novel view synthesis, though the results highlight significant challenges in achieving competitive performance. While the demonstrated FiLM-based camera conditioning and hybrid adapter architectures offer computational efficiency benefits, the substantial quality gaps compared to established methods—largely attributable to the 40× smaller training dataset—indicate that efficiency gains are insufficient to compensate for limited training data scale in practical novel view synthesis applications. The systematic experimental framework and performance evaluation provide valuable insights for the research community, particularly in understanding the limitations of adapter-based approaches and the continued need for more sophisticated architectures or training strategies to achieve state-of-the-art synthesis quality.

Bibliography

[1] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, Mar. 2003.

[3] Z. Chong, X. Dong, H. Li, S. Zhang, W. Zhang, X. Zhang, H. Zhao, and X. Liang. Catvton: Concatenation is all you need for virtual try-on with diffusion models, 2024.

[4] B. O. Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[5] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, E. VanderBilt, A. Kembhavi, C. Vondrick, G. Gkioxari, K. Ehsani, L. Schmidt, and A. Farhadi. Objaverse-xl: A universe of 10m+ 3d objects, 2023.

[6] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse: A universe of annotated 3d objects. *arXiv preprint arXiv:2212.08051*, 2022.

[7] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items, 2022.

[8] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, D. Podell, T. Dockhorn, Z. English, K. Lacey, A. Goodwin, Y. Marek, and R. Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024.

[9] R. Gao, A. Holynski, P. Henzler, A. Brussee, R. Martin-Brualla, P. Srinivasan, J. T. Barron, and B. Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024.

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.

[11] T. Hang, S. Gu, C. Li, J. Bao, D. Chen, H. Hu, X. Geng, and B. Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7441–7451, October 2023.

[12] J. Hessel, A. Holtzman, M. Forbes, R. L. Bras, and Y. Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022.

[13] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc.

[14] Z. Huang, Y. Guo, H. Wang, R. Yi, L. Ma, Y.-P. Cao, and L. Sheng. Mv-adapter: Multi-view consistent image generation made easy. *arXiv preprint arXiv:2412.03632*, 2024.

[15] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution, 2016.

[16] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2022.

[17] A. Lemay, C. Gros, O. Vincent, Y. Liu, J. Cohen, and J. Cohen-Adad. Benefits of linear conditioning for segmentation using metadata. 02 2021.

[18] T. Lindeberg. Scale invariant feature transform. *Scholarpedia*, 7, 05 2012.

[19] R. Liu, R. Wu, B. V. Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023.

[20] I. Loshchilov and F. Hutter. Decoupled weight decay regularization, 2019.

[21] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020.

[22] W. Peebles and S. Xie. Scalable diffusion models with transformers, 2023.

[23] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer, 2017.

[24] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.

[25] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[26] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[27] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: an efficient alternative to sift or surf. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2564–2571, 11 2011.

[28] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. 06 2016.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.

[31] P. Wang and Y. Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023.

[32] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[33] D. Watson, W. Chan, R. Martin-Brualla, J. Ho, A. Tagliasacchi, and M. Norouzi. Novel view synthesis with diffusion models, 2022.

[34] J. Xu, W. Cheng, Y. Gao, X. Wang, S. Gao, and Y. Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.

[35] Y. Xue, X. Xie, R. Marin, and G. Pons-Moll. Human 3diffusion: Realistic avatar creation via explicit 3d consistent diffusion models. *Arxiv*, 2024.

[36] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. 2023.

[37] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[38] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

List of Figures

List of Tables