Shading Concerned Generative Models for Fine-grained Photo-realistic Image Generation

Zhuo He

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Computing Science
College of Science and Engineering
University of Glasgow

June 2026

Acknowledgements

I would like to express my heartfelt gratitude to my supervisors Dr. Nicolas Pugeault and Dr. Paul Henderson for their invaluable guidance, continuous support, and constant encouragement throughout my academic journey. Their profound knowledge, insightful advice, and patient mentorship have greatly contributed to the completion of this thesis. I am deeply thankful for their dedication, constructive discussions, and unwavering belief in my abilities.

I would also like to sincerely thank my friends and colleagues Yingdong Ru, Eliyas Sulaiman, Lipeng Zhuang, Shiyu Fan for their support, encouragement, and companionship throughout this journey. Their insightful discussions, helpful suggestions, and constant motivation have made both my academic and personal experiences more rewarding.

Finally, I am eternally grateful to my parents Chunyuan He, Xia Cao for their unconditional love, endless support, and absolute confidence in me. Their sacrifices, encouragement, and understanding have been the foundation of my strength and perseverance. Without their continual care and faith, this achievement would not have been possible.

Abstract

Modern generative models can synthesize highly realistic images, but their internal representation of content, material, illumination, and viewpoint is often implicit. This makes the generated result difficult to analyse and edit in a controlled manner. The central question of this thesis is therefore how to introduce fine-grained control into generative models without sacrificing visual fidelity.

This thesis studies that question through a sequence of methods that progressively connect controllable generation with physics-based rendering. First, we investigate controllability inside a style-based generator and introduce *generative fields* as a way to analyse the spatial extent of channel-wise control in StyleGAN2. This analysis is used to relate different generator layers to coarse and fine facial attributes, and motivates a reference-guided editing framework that preserves identity while transferring pose and expression from a reference image. The resulting study shows that controllability inside a latent generator can be made more explicit when the spatial role of intermediate channels is analysed rather than treated as a purely implicit property of the model.

Second, we study illumination control as a rendering problem and propose a *physics-based neural deferred shading pipeline* for real-world portrait images. The method takes estimated material, geometry, and illumination attributes as input, and learns a scene-agnostic mapping from geomertry buffer representations to photorealistic shading under HDR environment lighting. To support this study, we construct FFHQ256-PBR, a large facial dataset with estimated PBR textures, geometry buffers, lighting, and camera parameters. This part of the thesis shows that a learned shader can compensate for the mismatch between inverse-rendered real-world attributes and classical rendering assumptions, making relighting more controllable in practice.

Third, we integrate this rendering perspective into text-to-image generation. We introduce *ShadingFusion*, a rendering-aware diffusion pipeline in which a modified latent diffusion model predicts decomposed material representations rather than only final RGB appearance, and a neural shader then renders the result under controllable illumination. To enable this setting, we construct a paired dataset of portrait images, estimated PBR attributes, and structured text descriptions, and we redesign the VAE decoder as a multi-head architecture that jointly reconstructs RGB content and G-buffer outputs. This

decomposition allows diffusion-based synthesis to retain photorealism while supporting explicit relighting and material-aware control after generation.

Finally, we extend the same idea to 3D generation. We propose *DiffGSPBR*, which combines generative 3D Gaussian splatting with deferred shading to produce decomposed 3D scenes whose materials, lighting, and viewpoint can be edited after generation. Built on top of a generative Gaussian-splatting backbone, the method adds a material-estimation head, a global illumination-estimation head, and a physics-based Gaussian deferred renderer that closes the loop through self-supervised reconstruction. In this way, the thesis moves from controllable 2D generation to editable 3D scene synthesis, showing that rendering-aware decomposition can support relightable and view-consistent generation in a unified framework.

Taken together, these contributions show that separating content generation from image formation is a practical way to improve the interpretability and controllability of generative models, rather than treating rendering as an implicit by-product of sampling. This thesis demonstrates that rendering can be brought back into the generative pipeline as an explicit and editable component for both 2D and 3D synthesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Image generation and image rendering have traditionally developed as different research directions. Generative models focus on synthesising novel content from data distributions, whereas rendering focuses on forming an image from an explicit scene description, material model, lighting configuration, and camera. In modern deep generative systems these two roles are often merged: a model directly predicts final RGB output, but the intermediate representation of geometry, material, and illumination remains implicit. As a result, the generated image may be visually convincing while still being difficult to analyse, edit, or relight in a controlled manner.

This thesis is motivated by the observation that fine-grained controllability requires a clearer separation between *what* is generated and *how* it is imaged. In particular, many practical editing tasks require some aspects of a scene to remain stable while others change in a physically meaningful way. Examples include preserving identity while changing pose, preserving content while changing illumination, or preserving structure while changing viewpoint. These requirements are common and well studied in computer graphics, but they are not naturally enforced in standard generative pipelines.

To clarify the discussion, this thesis distinguishes two coupled but conceptually different stages:

- **Content generation** is the synthesis of latent or structured representations that specify what should be present in the scene.
- **Rendering** is the image-formation process that maps those representations to observable output under geometry, material, lighting, and viewpoint constraints.

Figure 1.1 highlights the contrast between these two paradigms. Graphics-based rendering provides precise control over the image-formation process through explicit geometry, material, lighting, and camera parameters. However, it does not by itself solve the problem of generating new content, and the final appearance is still constrained by the assumptions and approximations of the rendering algorithm. Contemporary generative

Figure 1.1: Comparison of offline rendering, real-time rendering (Luckmouse, 2016) and the image generated from a statistical generative model

models, by contrast, can learn directly from real image distributions and therefore produce highly realistic results, but they typically do so without exposing the underlying factors needed for fine-grained control. The central challenge addressed in this thesis is therefore how to combine these complementary strengths: retaining the realism of learning-based synthesis while recovering some of the transparency and editability that are natural in graphics.

1.1 Motivation

Recent generative models achieve impressive perceptual quality, but fine-grained control over specific attributes remains challenging due to three fundamental limitations:

- **Feature entanglement and identity preservation.** Latent representations often mix pose, expression, texture, illumination, and identity [71, 91, 181]. As a result, changing one attribute can unintentionally modify others.
- **Weak physical interpretability.** Standard RGB generation does not expose the intermediate variables that are needed for relighting, material editing, or viewpoint control. This makes physically meaningful editing difficult even when the generated image looks realistic.
- **Limited supervision for controllable editing.** Many desired editing operations lack paired training data. Controllable generation must therefore rely on weaker forms of supervision, including reconstruction objectives, decomposition constraints, or learned intermediate representations that make the desired factors easier to isolate.

This thesis addresses these challenges through a staged progression. Chapter 3 studies fine-grained control inside an existing generator. Chapter 4 learns a rendering function for

real-world relighting. Chapters 5 and 6 then integrate decomposed rendering into 2D and 3D generative pipelines. The overall direction is therefore not a single all-in-one system, but a sequence of steps toward editable, physically informed generative models.

This progression matters because not all forms of control place the same demands on a model. Prompting a text-to-image model with the word "red" is a form of control, but it is not the same as requiring a portrait to preserve identity under pose changes, or requiring a relit result to remain consistent with the same material and geometry under a new environment map. The thesis therefore treats controllability as a hierarchy. At the weaker end, the user nudges the output distribution toward a semantic attribute. At the stronger end, the user intervenes on a meaningful variable while expecting the rest of the scene to remain stable. The methods developed later in the thesis target this stronger notion.

The motivation is also practical. Many applications do not require unrestricted image synthesis; they require dependable editing under constraints. Portrait editing for media production, digital humans, relighting, avatar creation, and content authoring all involve variables that users expect to manipulate separately. This is also why the thesis repeatedly returns to portrait data. Faces provide a demanding test case because identity, expression, skin reflectance, and shadow placement are all visually sensitive. If a model can preserve identity while changing pose, or preserve facial appearance while relighting under new conditions, then the underlying representation is likely capturing meaningful structure rather than only plausible texture.

1.2 Research Statement

The overall problem considered in this thesis can be stated as follows. Given a generative model that produces realistic images or 3D scenes, how should the internal representation and image-formation process be organized so that specific scene attributes can be modified intentionally while other attributes remain stable? The emphasis here is not only on whether a model can occasionally produce a desired edit, but on whether the representation supports a predictable correspondence between user intent, intermediate variables, and output appearance.

In current generative models, final RGB appearance is often predicted directly while many important control variables remain implicit. This coupling makes targeted editing difficult, especially when the desired change should preserve other properties of the scene. For example, a user may wish to change pose while preserving identity, relight a face while preserving material appearance, or alter viewpoint while preserving both content and illumination structure. This problem sits at the intersection of two traditions that have historically emphasised different priorities. Statistical generative modeling aims to fit rich

data distributions and synthesise realistic samples, often tolerating substantial internal entanglement as long as perceptual quality remains high. Computer graphics, by contrast, is built around explicit scene variables and deterministic rendering rules, even when the scene itself must be estimated from data. This thesis does not argue that one tradition should replace the other. Instead, it treats them as complementary: learned generative priors are valuable because they capture realism beyond hand-crafted graphics models, while rendering structure is valuable because it makes editing and reasoning possible.

This thesis therefore studies how ideas from rendering can be used to make generative models more editable while still preserving the flexibility of learned synthesis. The research strategy of this thesis is divided into three stages. First, study how content controllability can be introduced into a generative model at the representation level. Second, study how a physics-based shading process can be learned for real-world data. Third, integrate explicit rendering components into generative pipelines so that content, illumination, and viewpoint can be controlled more directly through corresponding modules. The resulting chapters therefore form a progression from controllable latent manipulation, through learned rendering, to decomposed 2D and 3D generation.

At the thesis level, this strategy can be understood as a move from *implicit control* to *explicit control*. Early in the thesis, control is recovered from a pretrained generator by identifying where interventions should enter its internal representation. Later, the image-formation process itself becomes an explicit, trainable module. In the final stage, generation is reorganized so that the model predicts structured intermediate representations before rendering the final output.

1.3 Research Questions

These stages are formalised by the following research questions:

- **RQ1:** How can latent or intermediate representations in generative models be used to achieve fine-grained control over synthesized content?
  - **RQ1.1:** Which parts of a generator control global and local attributes of the synthesized result?
  - **RQ1.2:** Can such control be made explicit enough to support later integration with rendering-based editing?
- **RQ2:** How can the shading process be learned for real-world images so that illumination becomes an explicit and controllable component?
  - **RQ2.1:** How should material, geometry, and illumination be represented for neural deferred rendering?

  - **RQ2.2:** Can a learned shading model generalise across real and synthetic domains well enough to support later generative use?

- **RQ3:** How can rendering-aware intermediate representations be integrated into generative models to produce editable 2D and 3D outputs?
  - **RQ3.1:** How should content generation and image formation be separated in a text-to-image generative pipeline?
  - **RQ3.2:** Can a 3D Gaussian representation be combined with deferred shading to support controllable material, illumination, and viewpoint editing?

1.4 Contributions

The thesis makes four main contributions.

- **Fine-grained control in a style-based generator.** Chapter 3 studies how different parts of StyleGAN contribute to controllable synthesis and introduces generative fields as an analysis tool for channel-wise control. Based on this analysis, it develops a reference-guided editing method for pose and expression control.
- **A learned rendering pipeline for real-world relighting.** Chapter 4 introduces a physics-based neural deferred shading framework that predicts photorealistic shading from estimated material, geometry, and illumination cues. It also introduces FFHQ256-PBR to support training and evaluation in the facial domain.
- **Rendering-aware text-to-image generation.** Chapter 5 proposes ShadingFusion, a two-stage pipeline that generates decomposed material representations from text and renders them with a neural shader, enabling more explicit control over material and illumination than direct RGB synthesis.
- **Rendering-aware 3D generative synthesis.** Chapter 6 extends the same principle to 3D Gaussian splatting by introducing a pipeline that predicts editable 3D representations together with material and illumination components, allowing relightable 3D generation from text or image conditions.

1.5 Thesis Outline

The remainder of this thesis is organised as follows. Although the chapters address different technical settings, they are intended to be read as a connected progression rather than as isolated projects. The thesis begins by clarifying the background needed to discuss

controllable generation and rendering-aware modeling. It then moves from controllability within an existing image generator, to learned shading for real-world relighting, and finally to 2D and 3D generative pipelines in which rendering is treated as an explicit part of the synthesis process. The outline below therefore serves not only as a chapter guide, but also as a map of how the overall argument develops.

- Chapter 2 reviews the literature needed to support the thesis, with particular emphasis on controllable generation, rendering, inverse rendering, and related decomposition methods. Rather than serving as a general introduction to machine learning, it selectively introduces the probabilistic, generative, and graphics concepts that are used later in the thesis, thereby establishing the conceptual link between latent-variable generation and explicit image formation. The chapter first summarizes only those probability and learning concepts that recur later, such as conditional generation, latent variables, and task-dependent evaluation metrics. It then reviews the main generative model families used throughout the thesis, including VAEs, GANs, flow-based models, and diffusion models, focusing on the representational properties that matter for controllability. Finally, it surveys rendering, inverse rendering, deferred shading, and 3D reconstruction, which together provide the graphics foundation for the later chapters. The role of Chapter 2 is therefore to narrow the background to the concepts that the thesis relies on, while also making clear why controllable generation and rendering should be studied together rather than as separate topics.

- Chapter 3 addresses RQ1 by analysing controllability inside StyleGAN and proposing a reference-guided editing framework for pose and expression control. It introduces the notion of *generative fields* to characterize the spatial scale influenced by different generator layers, and uses this analysis to explain where control should be injected in style space to preserve identity while transferring pose and expression. The practical task studied in this chapter is reference-guided face editing: given a generated identity image and a reference image providing pose and expression cues, the goal is to produce an output that preserves identity while following the reference attributes. The chapter therefore combines two contributions. The first is analytical: it develops a quantitative account of how different layers in StyleGAN correspond to different scales of image control. The second is methodological: it uses this analysis to build a style-space editing pipeline that injects control signals at appropriate locations. The experiments then examine both editing quality and the sparsity of the control signal, showing how latent controllability can be made more interpretable when the generator is studied at the level of intermediate representations.

- Chapter 4 addresses RQ2 by introducing a physics-based neural deferred shading

pipeline for controllable relighting of real-world images and by establishing the rendering component used later in the thesis. The chapter formulates relighting as a scene-agnostic neural shading problem, constructs the FFHQ256-PBR dataset with estimated material and illumination attributes, and studies how a learned shader can map G-buffer representations to photorealistic appearance under novel environment lighting. More specifically, the chapter asks whether the shading process can be learned directly from estimated material, geometry, and illumination cues, even when those cues are noisy and are not produced under ideal graphics assumptions. To answer that question, it develops a two-stage neural shader consisting of a physics-informed deferred shading model and a learned shadow estimator, and trains the system on large-scale estimated supervision. This chapter is important not only because it addresses relighting in its own right, but also because it establishes the rendering formulation reused in the later generative chapters. In other words, Chapter 4 provides the bridge from latent controllability toward rendering-aware generation by showing that shading can itself be learned as an explicit and controllable component.

- Chapter 5 addresses the 2D part of RQ3 by combining diffusion-based generation with decomposed material prediction and neural rendering for controllable text-to-image synthesis. It introduces ShadingFusion, a rendering-aware diffusion pipeline built around a multi-head VAE decoder, a text-conditioned latent denoiser, and a neural shader, together with a dataset construction pipeline that pairs portrait images with estimated PBR attributes and structured text descriptions for relightable generation. The central idea of this chapter is that a text-to-image model should not be asked to predict only final RGB appearance if the desired goal is later editing under new lighting. Instead, the model is redesigned so that generation and rendering are separated: the diffusion backbone predicts decomposed material representations, and a dedicated renderer produces the final image under specified illumination. To support that design, the chapter introduces CelebA-PBR-Text, a dataset that combines portrait images, estimated inverse-rendering outputs, and structured prompt information collected through an MLLM-based pipeline. The chapter then evaluates generation, relighting, and ablation settings, showing that rendering-aware decomposition can improve editability while remaining competitive in perceptual image quality. In the thesis as a whole, Chapter 5 is the first point at which the rendering component is integrated directly into a generative model.

- Chapter 6 addresses the 3D part of RQ3 by integrating 3D Gaussian splatting with deferred shading to support controllable generation, relighting, and viewpoint manipulation. Building on a generative Gaussian-splatting backbone, it introduces material and illumination estimation heads together with a physics-based Gaussian

deferred renderer, so that the generated 3D scene can be decomposed into editable geometry, material, and lighting components. Relative to Chapter 5, the key change here is that the problem is no longer limited to a single relightable image plane; instead, the objective is to synthesize a view-consistent 3D scene that can be rendered from novel viewpoints under changed illumination. The chapter therefore decomposes the pipeline into three stages: Gaussian scene generation, scene-level material and illumination estimation, and deferred rendering with direct and indirect lighting. It evaluates both text-conditioned and image-conditioned generation, and also examines relighting of the resulting decomposed scenes. This chapter represents the most complete version of the thesis argument, because it combines generative modeling, decomposition, rendering, and post-generation editability within one explicit 3D representation.

- Chapter 7 summarises the contributions of the thesis, revisits the research questions, discusses limitations, and outlines directions for future work. It also reflects on the broader thesis argument that controllability improves when generation and rendering are separated more explicitly, and identifies the remaining challenges in extending this principle to richer scenes and more general settings. Rather than functioning only as a summary, the chapter brings together the four technical contributions and evaluates them against the thesis-level claims made in the introduction. It discusses what has been achieved at the levels of latent controllability, learned shading, 2D rendering-aware generation, and 3D editable synthesis, and it makes explicit the limitations that remain in each setting. The chapter finally outlines several directions for future work, including stronger real-world material modeling, more complex scenes, real-time performance, and richer user-facing editing interfaces. In this sense, Chapter 7 closes the thesis by returning from the individual methods to the broader question of how explicit rendering structure can improve controllable generative models.

1.6 Publications

The work presented in this thesis relates to the following works. First is the peer-reviewed publication while the second represents papers that are under review:

1. He, Z., Henderson, P., Pugeault, N. (2026). Beyond Reconstruction: A Physics Based Neural Deferred Shader for Photo-Realistic Rendering. Artificial Neural Networks and Machine Learning – ICANN 2025, 378–389. Springer Nature Switzerland.

In addition to the publications listed above, the following enumerated list represents papers that are under peer-review:

1. He, Z., Henderson, P., Pugeault, N. (2025). Generative Fields: Uncovering Hierarchical Feature Control for StyleGAN via Inverted Receptive Fields In ArXiv Preprint.

2. He, Z., Henderson, P., Pugeault, N. (2023). ShadingFusion: Decomposing Diffusion Models for Physically-Grounded Image Synthesis and Relighting. In ArXiv Preprint.

3. He, Z., Henderson, P., Pugeault, N. (2025). DiffGSPBR: physics-Based Rendering of Generative Gaussian Splats for Decomposed 3D Synthesis. In ArXiv Preprint.

Chapter 2

Background

2.1 Introduction

This chapter reviews the background that is directly needed by the thesis. It is not intended to be a general introduction to machine learning or probability theory; rather, it highlights the concepts and areas of prior work that are most relevant to controllable generation, neural rendering, relighting, and decomposed 2D and 3D synthesis. In particular, the later chapters rely on three strands of background: selected probabilistic concepts used in generative modeling, the main families of deep generative models, and rendering and reconstruction methods from computer graphics.

1. Section 2.2 summarises selected probability and learning concepts that are directly used later in the thesis. These summaries are intentionally selective and are included only where they support the notation or methodological choices of later chapters.

2. Section 2.3 reviews the main families of deep generative models used throughout the thesis, with emphasis on the representational and training properties that matter for controllability.

3. Section 2.4 reviews rendering, inverse rendering, deferred shading, and reconstruction methods that motivate the rendering-aware contributions in later chapters.

2.2 Selected Probability and Learning Concepts

This section briefly reviews the concepts from probability and learning theory that recur in later chapters. The presentation is intentionally selective: it is not meant to cover the full scope of these subjects, but only the parts that are repeatedly used in the formulation of generative models, neural rendering losses, and evaluation protocols.

2.2.1 Probability, Conditioning, and Latent Variables

Generative models aim to represent a data distribution $p(x)$, or a conditional distribution $p(x \mid c)$ when external control signals are provided. The distinction between joint, marginal, and conditional distributions is therefore central to this thesis. In particular, the later chapters rely on conditional generation, where the output must depend on structured inputs such as pose, text, material cues, or lighting.

Latent-variable models additionally introduce hidden variables z and factorise the joint distribution as

$$p(x, z) = p(x \mid z)\, p(z).$$

This perspective is especially relevant for variational autoencoders, diffusion models operating in latent space, and style-based generators whose intermediate representations are manipulated for controllability.

Gaussian distributions. Gaussian distributions play a central role in modern generative modeling. A univariate Gaussian is written as

$$p(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right),$$

and its multivariate form extends naturally to vectors with covariance structure. In later chapters, Gaussian priors and Gaussian noise appear in VAE latent spaces, diffusion forward processes, and sampling procedures.

2.2.2 Entropy and Divergence

The thesis uses only a small subset of information-theoretic concepts. The most relevant one is Shannon entropy, which measures uncertainty in a distribution:

$$H(X) = -\sum_{i=1}^{n} P(x_i)\, \log P(x_i).$$

Rather than surveying information theory broadly, the main practical need here is to motivate divergence measures used in training and analysis.

Kullback-Leibler divergence. The Kullback-Leibler divergence between distributions P and Q is

$$D_{\mathrm{KL}}(P \,\|\, Q) = \sum_{x} P(x)\, \log \frac{P(x)}{Q(x)}.$$

It appears directly in the VAE objective and more generally measures how one distribution differs from another.

Other divergence families. Beyond KL divergence, later discussion occasionally refers to Jensen-Shannon, Wasserstein, or related discrepancy measures. These are relevant because different generative models optimise different statistical objectives, which affects both training behaviour and controllability.

2.2.3 Learning Setup, Model Families, and Metrics

The methods in this thesis use a mixture of supervised, self-supervised, and weakly supervised learning. For that reason, only a limited subset of standard machine learning concepts needs to be stated explicitly here.

Prediction heads and activation functions. Simple linear prediction can be written as

$$\hat{y}_i = \boldsymbol{\theta}^\top \mathbf{x}_i + b \tag{2.1}$$

with parameters $\boldsymbol{\theta} \in \mathbb{R}^d$ and bias $b \in \mathbb{R}$. In classification settings, the output may then be passed through a link function such as the sigmoid

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

or the softmax

$$\operatorname{softmax}(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{N} e^{z_k}}.$$

These equations are included only as standard examples of output heads, not as an exhaustive description of classifier design.

Evaluation metrics. Likewise, classification metrics such as precision, recall, and F_1 are included as standard examples for supervised prediction tasks, while regression or reconstruction metrics such as MSE, PSNR, and SSIM are included because they recur in later experiments. Their appearance here should be read as task-dependent examples rather than as a universal metric taxonomy for machine learning.

Neural architectures. The later chapters rely mainly on three classes of neural modules: MLPs for point-wise or per-sample prediction, CNNs for spatial feature extraction and decoding, and transformer-style attention blocks for global context aggregation. Back-propagation and gradient-based optimisation remain the standard training mechanism throughout the thesis and are assumed in the descriptions of all later methods.

2.2.4 Conditioning, Supervision, and Controllability

Because this thesis is concerned with controllable generation rather than unconditional sampling alone, it is useful to distinguish several ways in which a model may be conditioned. The most direct case is *explicit conditional generation*, where the model is trained to represent a distribution $p(x \mid c)$ and the condition c is given in a structured form such as text, pose, geometry, or lighting. This is the setting used by many conditional GANs and text-conditioned diffusion models.

However, useful control can also be recovered from the internal representation of a pretrained model. Style-based generators, for example, expose intermediate spaces whose channels can be manipulated after training [71, 162]. This distinction matters for the thesis because Chapter 3 relies on post hoc analysis of internal representations, whereas Chapters 5 and 6 rely more heavily on explicit conditioning and structured prediction heads.

Conditioning alone does not guarantee selective control. A model may respond to a condition while still entangling the requested factor with many others. The thesis therefore treats controllability as stronger than mere conditional dependence: the intended factor should change when asked, while unrelated factors remain stable within the limits of the task. This idea is related to disentanglement, but more practical in scope. Rather than requiring a fully factorised latent space, many useful systems aim for *task-aligned disentanglement*, where only the factors needed for editing are separated clearly enough to be manipulated.

Supervision strategy is central to whether such control can be learned. Fully supervised labels are often unavailable for physical attributes such as roughness, environment lighting, or scene structure. The methods in this thesis therefore combine paired image constraints, estimated inverse-rendering attributes, reconstruction objectives, and explicit rendering interfaces. This also motivates a final distinction between *conditioning signals* and *control interfaces*. A model may consume rich conditions internally, but only some of them remain meaningful and editable after training. The later chapters repeatedly favour representations such as albedo maps, G-buffers, or scene-level illumination codes because they can function as reusable interfaces rather than hidden pathways.

2.2.5 Loss Design and Multi-criteria Evaluation

The learning problems studied in this thesis rarely optimise a single objective. A model may need to reconstruct images, preserve identity, match a target distribution, remain plausible under relighting, and expose intermediate variables that can still be edited afterward. These requirements are only partially aligned, so training often combines several losses that each proxy a different aspect of the desired behaviour.

Pixel-wise losses such as MSE or L_1 distance help stabilise dense prediction, but by themselves they often favour overly smooth outputs. Perceptual or adversarial objectives improve realism, yet they may encourage shortcuts that weaken the intended decomposition. Distributional metrics such as FID are informative about sample quality at the set level, but they say little about whether an edit preserves identity or whether a relighting operation behaves consistently. The background lesson is therefore simple: evaluation must follow the representation. If a method exposes geometry, material, and illumination explicitly, then those variables should also be evaluated through downstream manipulation rather than only through final RGB quality.

This multi-criteria setting also changes how ablations should be interpreted. Removing a rendering branch, weakening the conditioning interface, or replacing a structured head with direct RGB prediction may improve one metric while damaging controllability. In the later chapters, ablations are therefore used not only to compare components, but also to test where the control signal actually resides in the pipeline.

2.3 Statistical Generative Models

Generative modeling seeks to learn a distribution over high-dimensional data such as images or 3D scenes and to sample novel instances from that distribution. For the purposes of this thesis, the most relevant model families are those that expose useful intermediate representations, support controllable conditioning, or can be integrated with rendering-aware structure. We therefore review four representative classes: variational autoencoders, generative adversarial networks, flow-based models, and diffusion models.

2.3.1 Variational Autoencoders

Variational Autoencoders (VAEs) [80] posit a latent variable model

$$p_\theta(x, z) = p_\theta(x \mid z)\, p(z),$$

where $z \in \mathbb{R}^d$ is drawn from a simple prior $p(z)$ (typically $\mathcal{N}(0, I)$). Exact maximum-likelihood estimation $\max_\theta \log p_\theta(x)$ is intractable due to the integral $\log p_\theta(x) = \log \int p_\theta(x \mid z)p(z)\, dz$. Instead, VAEs introduce an approximate posterior $q_\phi(z \mid x)$ (the encoder) and optimize the Evidence Lower Bound (ELBO):

$$\mathcal{L}_{\text{ELBO}}(\theta, \phi\,;\, x) = \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x \mid z)] - D_{\text{KL}}\big(q_\phi(z \mid x) \,\|\, p(z)\big).$$

The first term encourages reconstruction accuracy, while the KL term regularizes q_ϕ toward the prior. To enable gradient-based learning through the stochastic sampling of z, one uses

the reparameterization trick:

$$z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I),$$

so that gradients w.r.t. ϕ back-propagate through μ_ϕ, σ_ϕ. After training, new samples are generated by sampling $z \sim p(z)$ and decoding via $p_\theta(x \mid z)$.

2.3.2 Generative Adversarial Networks

Generative Adversarial Networks (GANs) [36] formulate generation as a two-player minimax game between a generator $G_\theta(z)$ and a discriminator $D_\psi(x)$. The generator transforms noise $z \sim p(z)$ into synthetic data $G_\theta(z)$, while the discriminator outputs $D_\psi(x) \in (0, 1)$, the estimated probability that x is real. The adversarial objective is

$$\min_\theta \max_\psi \; V(\theta, \psi) = \mathbb{E}_{x \sim p_{\text{data}}}[\log D_\psi(x)] + \mathbb{E}_{z \sim p(z)}[\log\big(1 - D_\psi(G_\theta(z))\big)].$$

At the Nash equilibrium, G_θ reproduces the data distribution exactly, and $D_\psi(x) = \frac{1}{2}$. In practice, one alternates gradient-descent steps on ψ (to improve discrimination) and on θ (to better fool D). Numerous variants address instability (e.g., Wasserstein GANs [3]) by replacing the loss or adding gradient penalties.

2.3.3 Flow-based Models

Flow-based models [126] learn an explicit density via a sequence of invertible, differentiable transformations f_i. Starting from $z_0 \sim p(z_0)$ (often Gaussian), one sets

$$z_i = f_i(z_{i-1}), \quad i = 1, \ldots, K, \quad x = z_K.$$

By the change-of-variables formula,

$$\log p_X(x) = \log p_Z(z_0) - \sum_{i=1}^{K} \log\Big|\det \tfrac{\partial f_i}{\partial z_{i-1}}\Big|.$$

Provided each f_i has a tractable Jacobian determinant (e.g., coupling layers in RealNVP [25]), one can maximize $\log p_X(x)$ exactly via gradient descent. Sampling is equally straightforward: draw z_0 and apply the forward flow.

2.3.4 Diffusion Models

Diffusion models [143, 52] define generation as the reversal of a fixed forward noising process. The forward diffusion slowly corrupts data $x_0 \sim p_{\text{data}}$ with Gaussian noise:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\big(x_t; \sqrt{1-\beta_t}\, x_{t-1},\, \beta_t I\big), \quad t = 1, \ldots, T,$$

where $\{\beta_t\}$ is a schedule of variances. The reverse process is parameterized by a neural network $\epsilon_\theta(x_t, t)$ predicting the noise component, yielding

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\big(x_{t-1}; \tfrac{1}{\sqrt{1-\beta_t}}\big(x_t - \beta_t\, \epsilon_\theta(x_t, t)\big),\, \tilde{\beta}_t I\big).$$

Training minimizes a simple mean-squared error between true noise and ϵ_θ:

$$\mathbb{E}_{x_0, \epsilon, t}\Big[\big\|\epsilon - \epsilon_\theta(x_t, t)\big\|^2\Big].$$

Sampling begins at $x_T \sim \mathcal{N}(0, I)$ and iteratively applies the learned reverse transitions to produce a clean sample x_0. Diffusion models have achieved state-of-the-art fidelity in high-resolution image synthesis and 3D generative tasks.

Latent Diffusion and Stable Diffusion

While the above describes pixel-space diffusion, *latent diffusion models* perform the diffusion process in a lower-dimensional latent space, yielding substantial efficiency gains. Stable Diffusion [128] first encodes an image x with a pretrained variational autoencoder (VAE):

$$z = E(x), \quad x \approx D(z),$$

where E and D are the VAE's encoder and decoder. The forward and reverse noising processes then operate on $z \in \mathbb{R}^d$ rather than on pixels. A U-Net $\epsilon_\theta(z_t, t, c)$ predicts noise conditioned on textual embeddings c (e.g., from a CLIP or transformer text encoder). The training loss becomes

$$\mathbb{E}_{z_0, \epsilon, t, c}\Big[\big\|\epsilon - \epsilon_\theta(z_t, t, c)\big\|^2\Big],$$

where $z_t = \sqrt{\bar{\alpha}_t}\, z_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon$. At sampling time, one draws $z_T \sim \mathcal{N}(0, I)$ and iteratively applies the reverse transitions to obtain z_0, then decodes $x = D(z_0)$. Conditioning on c via cross-attention layers enables fine-grained control over the generated content.

This latent formulation drastically reduces memory and compute requirements compared to pixel-space diffusion, enabling high-resolution synthesis during inference.

2.3.5 Controllability in Generative Models

The model families above differ not only in likelihood formulation or sampling procedure, but also in the kinds of control interfaces they naturally support. This is particularly important for the present thesis because controllability is not treated as an optional add-on. It is one of the main criteria by which a generative representation is judged.

In VAEs, control is often introduced through structured latent variables, conditional priors, or disentangling regularizers. Their encoder-decoder structure makes them attractive when one wants an explicit bottleneck that can be interpreted or manipulated, but the same bottleneck can also limit fidelity if it is too restrictive. GANs historically offered stronger image quality, and much work on controllability therefore focused on latent traversals, factorised latent spaces, attribute conditioning, and later style-based intermediate spaces [16, 71, 162]. A key lesson from that literature is that good controllability often depends less on the nominal model family than on where the controllable variable enters the synthesis process. Intervening at a global latent vector can yield coarse semantic change, whereas intervening in intermediate layers may support finer local control.

Flow-based models are less prominent in the later chapters of this thesis, but they remain conceptually relevant because they highlight an alternative ideal: exact invertibility between data and latent variables. In principle, invertibility is attractive for editing because every sample has a well-defined representation in latent space and vice versa. In practice, however, the restrictions needed for tractable Jacobians may limit the representational flexibility needed for large-scale image synthesis. Their importance here is mainly pedagogical: they clarify that explicit invertibility alone does not solve controllability unless the latent axes correspond to meaningful factors.

Diffusion models changed the control landscape substantially because they made it natural to condition synthesis on rich signals such as text, sketches, masks, depth maps, normals, or adapter features. In modern text-to-image systems, conditioning is often injected repeatedly through cross-attention or additional control branches, allowing the same backbone to be steered in many ways [132, 128, 184]. This flexibility is one reason diffusion models are central to Chapter 5 and part of the motivation for Chapter 6. Yet diffusion models also illustrate the difference between control and dependency. A model can be highly responsive to prompts and still fail to preserve identity, geometry, or material properties under targeted edits.

From the perspective of this thesis, controllability depends on three representation choices. First, there is the choice of *where* control enters: at the input, in the latent space, at intermediate feature maps, or at an explicit rendering stage. Second, there is the choice of *what* the control variable represents: semantic category, text, geometry, material, illumination, viewpoint, or some learned factor with only partial interpretation. Third, there is the choice of *how strongly* the rest of the pipeline is constrained to respect that

variable. An attribute that enters only as a weak conditioning signal may not survive the rest of the synthesis process if the model can explain the data more easily by entangling it with other factors.

These observations motivate a distinction between semantic controllability and structural controllability. Semantic controllability refers to the ability to steer the output toward a concept recognizable to humans, such as "smiling," "wearing glasses," or "studio lighting." Structural controllability refers to the ability to operate on a variable that participates in a larger image-formation model and therefore has predictable consequences when edited. The thesis increasingly shifts from the first to the second notion. Chapter 3 still operates near semantic controllability, because pose and expression are manipulated inside a style-based generator. Chapters 4–6 progressively move toward structural controllability by making geometry, material, illumination, and viewpoint explicit parts of the generation pipeline.

The literature also reveals a recurring trade-off between end-to-end flexibility and reusable intermediate structure. A direct conditional model can often learn very strong one-shot mappings from condition to image, especially when the condition and target distribution are tightly aligned. However, such models may offer limited reuse after inference: once the final RGB output is produced, the internal factors that generated it are not accessible for further editing. Rendering-aware and decomposition-aware models sacrifice some of this directness in order to create representations that remain useful after generation. This trade-off is central to the thesis and explains why later chapters repeatedly compare modular structured pipelines against stronger but less interpretable direct baselines.

Finally, controllability should be understood as a property of the whole pipeline rather than of the generator alone. A model may contain an apparently interpretable latent space, but if the decoder or renderer downstream amplifies entanglement, the final user-facing control remains weak. Conversely, a model may rely on a high-capacity generator internally yet still offer strong control if the interface between generation and rendering is well structured. This pipeline-level view is especially important for Chapters 5 and 6, where the practical control interface is not a single latent vector but the combination of decomposition heads, illumination controls, and rendering modules.

2.4 Rendering and Reconstruction

In this section, we review the core techniques in computer graphics that underpin our neural rendering pipeline: classical rendering via the rendering equation, differentiable rendering for gradient-based inverse problems, inverse rendering for scene parameter estimation, deferred shading for real-time rendering, and 3D reconstruction for recovering geometry from images.

The Rendering Equation. Physics-based rendering is founded on the rendering equation [64], which expresses the outgoing radiance L_o at a point x in direction ω_o as

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_\Omega f_r(x, \omega_i, \omega_o)\, L_i(x, \omega_i)\, \langle n, \omega_i \rangle\, d\omega_i,$$

where L_e is emitted radiance, L_i is incoming radiance from direction ω_i, f_r is the bidirectional reflectance distribution function (BRDF), n is the surface normal, and $\langle n, \omega_i \rangle$ is the cosine term. Monte Carlo integration approximates this integral via random sampling of light directions [119].

Differentiable Rendering. Differentiable renderers compute gradients of rendered images with respect to scene parameters θ (geometry, materials, lighting), enabling optimization via backpropagation. Early work such as OpenDR [102] and the MitsubaDR wrapper [88] propagate derivatives through rasterization and Monte Carlo estimators. More recent neural approaches approximate visibility and shading derivatives through soft rasterization [101] or explicit derivative computations in ray tracing [87].

Inverse Rendering. Inverse rendering seeks to recover scene attributes θ from one or more images I. Formulated as

$$\hat{\theta} = \arg\min_\theta \| R(\theta) - I \|^2 + \mathcal{R}(\theta),$$

where $R(\theta)$ is a differentiable renderer and $\mathcal{R}$ encodes priors (e.g., smoothness). Classical methods estimate reflectance and geometry separately [21, 124], while recent deep inverse renderers jointly learn both via neural networks and differentiable ray marchers [134, 7].

Deferred Shading. Deferred shading [22, 12] decouples geometry and lighting by first rendering a G-buffer containing per-pixel attributes (normals, albedo, depth) and then performing lighting in screen space:

$$L_o(p) = \sum_l f_r\big(n_p, \omega_{l \to p}, \omega_o\big)\, L_l\, V(p, l)\, \langle n_p, \omega_{l \to p} \rangle,$$

for each light l, where V is visibility. Deferred methods allow many lights at low cost and facilitate neural augmentations on G-buffers [85].

3D Reconstruction. Recovering geometry from images includes multi-view stereo (MVS) and more recent neural methods. Classical MVS [31] aligns depth maps via photometric consistency. Neural Radiance Fields (NeRF) [109] parameterize a continuous volumetric scene by an MLP $F_\theta(\mathbf{x}, \mathbf{d})$ that predicts density and view-dependent color, rendering images

via differentiable volume rendering:

$$\hat{C}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\,\sigma(\mathbf{x}(t))\,c(\mathbf{x}(t),\mathbf{d})\,dt, \quad T(t) = \exp\Big(-\int_{t_n}^{t} \sigma(\mathbf{x}(s))\,ds\Big).$$

NeRF variants extend this to dynamic scenes, unbounded domains, and fast neural splatting [75].

2.4.1 Intrinsic Decomposition and Relighting

The rendering and reconstruction topics above become especially relevant when the goal is not only to reproduce an observed image, but to change the illumination or material conditions after inference. In such cases, one often needs some form of *intrinsic decomposition*: an estimate of which parts of image appearance should be attributed to reflectance, shading, geometry, normals, roughness, metallicity, or lighting. Classical intrinsic image decomposition usually focused on separating reflectance from shading, but more recent work extends this idea toward richer physically based attributes and learned inverse-rendering pipelines [106, 92, 134].

This decomposition problem is difficult because the image formation process is fundamentally ambiguous. Different combinations of illumination, material, and geometry can produce very similar RGB observations. A perfectly white surface under colored lighting may resemble a colored surface under neutral lighting; a specular highlight can arise from material or environment changes; small geometry errors can be compensated by texture or shading changes in image space. These ambiguities explain why direct prediction of physical attributes from a single image is often noisy and why many practical systems combine data-driven priors with rendering constraints rather than relying on analytic inversion alone.

For controllable synthesis, however, imperfect decomposition can still be useful if it exposes the right *kind* of structure. The later chapters of this thesis do not require a universally correct inverse-rendering solution for every pixel. Instead, they require a decomposition that is stable enough to support meaningful downstream relighting and editing. This is one reason why the thesis repeatedly relies on G-buffer-like channels and PBR-inspired attributes. Even when those estimates are imperfect, they provide a more actionable interface than an opaque latent code because they can be perturbed in ways that correspond to lighting or material operations.

Relighting methods illustrate this point clearly. If the goal is to produce a single visually plausible relit image, an end-to-end conditional model may perform well without ever exposing explicit geometry or material maps. But if the goal is to support repeated edits, analysis of failure cases, or integration with later rendering modules, then explicit decomposition becomes far more valuable. The user can inspect whether the albedo

estimate is plausible, whether normals drift across facial regions, or whether lighting changes behave consistently under repeated operations. In other words, decomposition is not only a means toward better rendering; it is also a diagnostic tool for controllability.

This diagnostic role has become more prominent in recent rendering-aware generative systems. Methods such as RGB↔X treat material and lighting channels not merely as intermediate supervision targets but as reusable variables for downstream synthesis [179]. Likewise, unified inverse-and-forward rendering systems show that even noisy estimated attributes can still define a useful control space when coupled with a sufficiently strong learned renderer [93]. These developments reinforce a central thesis claim: the value of an intrinsic representation lies not only in whether it is physically exact, but in whether it supports reliable editing and rendering operations afterward.

From a methodological perspective, intrinsic decomposition also changes the nature of the training signal. Instead of comparing only final RGB outputs, one can supervise or regularize intermediate maps, constrain them through synthetic renderings, and examine how errors propagate through a renderer. This is important for the later chapters because the proposed methods frequently use estimated inverse-rendering outputs to bootstrap new generative pipelines. The decomposition is therefore both a source of supervision and a target of control.

2.4.2 Scene Representations for Editable 3D Generation

The 3D part of this thesis requires one further layer of background: the choice of scene representation. A representation suitable for controllable 3D generation must do more than render one plausible image. It should remain coherent across viewpoints, admit efficient optimisation or generation, and expose parameters that can be linked to geometry, material, and illumination. Different 3D representations satisfy these requirements to different degrees.

Classical graphics pipelines often represent a scene explicitly through meshes, textures, material parameters, and lights. These are highly editable, but difficult to infer automatically from sparse images or text prompts. Neural radiance fields relaxed some of these constraints by representing a scene implicitly through a continuous volumetric function, enabling remarkable view synthesis from photographs [109]. Yet the same implicitness that gives NeRF its flexibility also makes direct post hoc editing challenging. Geometry, appearance, and view-dependent effects are often entangled inside a shared radiance field, and rendering requires expensive sampling along rays.

The emergence of 3D Gaussian splatting changed this landscape by offering a representation that is both differentiable and more explicit at the scene-primitive level [75]. A Gaussian-based scene consists of a collection of primitives with location, covariance, opacity, and color or feature attributes. Compared with dense volumetric rendering, Gaussian

splats can be rendered efficiently and support direct manipulation of scene primitives. This makes them attractive not only for view synthesis but also for editable 3D generation, where efficiency and local control both matter.

However, a Gaussian scene becomes truly useful for controllable rendering only when appearance is decomposed beyond static color. Recent work therefore augments Gaussian splatting with shading functions, inverse-rendering objectives, or deferred rendering stages that separate geometry, visibility, and material effects [160, 15]. These developments are directly relevant to Chapter 6 because they show that Gaussian representations can serve as more than fast radiance fields. They can also become carriers of editable physical attributes such as normals, BRDF parameters, or relightable appearance codes.

The relevance to generative modeling is twofold. First, Gaussian representations are compatible with modern 3D generative backbones, which increasingly learn scene structure from 2D diffusion priors, multi-view consistency objectives, or direct Gaussian synthesis [148]. Second, they provide a practical midpoint between fully implicit radiance fields and fully explicit graphics assets. A Gaussian scene can be generated from learned priors, rendered efficiently, and then edited through explicit scene-level attributes. This combination aligns closely with the thesis goal of separating content generation from image formation without giving up generative flexibility.

From the perspective of controllability, the main advantage of these newer 3D scene representations is that they make viewpoint change and lighting change part of the same problem rather than separate post-processing steps. If a 3D representation stores only appearance from a limited set of views, then relighting or material editing becomes ill posed. If it stores geometry and material information more explicitly, then a rendering layer can recompute appearance under new conditions in a way that is at least partially constrained by scene structure. This is precisely the direction taken in the later thesis chapters: Chapter 5 introduces decomposition in image space, while Chapter 6 moves the same principle into a scene representation that remains valid under novel views.

The broader background lesson is that representation choice determines not only rendering quality but the *type* of control a model can support. Meshes favour explicit surface editing but are hard to infer; radiance fields favour fidelity but may hide controllable factors; Gaussian splats offer an intermediate level of explicitness with efficient rendering. The thesis builds on this emerging middle ground because it offers a realistic path toward editable 3D generation with physically meaningful intermediate variables.

2.4.3 Recent Rendering-aware Generative Models

Recent work has begun to connect three areas that were often treated separately in earlier literature: inverse rendering, forward rendering, and generative image synthesis. A common theme in these systems is that physically meaningful intermediate channels such as albedo,

roughness, normals, and depth are no longer viewed only as analysis outputs. Instead, they are treated as an explicit interface between scene understanding and controllable image synthesis. This trend is directly relevant to the present thesis because it supports the broader claim that editable intermediate structure can improve relighting, material editing, and compositional control without giving up the realism of learning-based generation.

RGB↔X. RGB↔X [179] provides a particularly relevant bridge between graphics and generative modeling. It formulates both RGB→X decomposition and X→RGB synthesis inside one material-aware and lighting-aware diffusion framework. In that setting, intrinsic channels such as albedo, roughness, metallicity, and lighting are not treated merely as supervision targets; they are treated as controllable inputs for subsequent synthesis. This is conceptually close to the central argument of this thesis. The main difference is scope: RGB↔X focuses on interior scenes in the image domain, whereas the present thesis moves from portrait relighting to portrait text-to-image generation with material decomposition and finally to decomposed 3D generation.

DiffusionRenderer. DiffusionRenderer [93] is a representative example of a unified inverse-and-forward rendering system built on diffusion priors. It first estimates G-buffer-style scene attributes from real-world videos and then synthesizes photorealistic images from those estimated attributes without explicitly simulating light transport. This matters because it treats G-buffers as a practical control space for downstream editing and shows that a learned renderer can still function when the scene attributes are noisy estimates rather than artist-authored ground truth. Relative to this thesis, DiffusionRenderer is closest in spirit to Chapter 4 and Chapter 5. The difference is that its emphasis is a unified video-based framework, whereas the present thesis studies a staged progression from relighting, to 2D generation with explicit decomposition, to 3D generation with explicit decomposition.

UniRelight. UniRelight [44] explores a different point in the design space. Instead of separating decomposition and synthesis into distinct stages, it learns albedo estimation and relit video synthesis jointly in a single diffusion-based model. This design prioritises end-to-end synthesis quality and temporal consistency, especially for video, by allowing the generative model to internalise more of the decomposition problem. It is therefore a useful contrast to the present thesis, which generally favours explicit intermediate representations because they are easier to inspect and edit. UniRelight represents the opposite trade-off: less modularity, but potentially stronger synthesis when the goal is one final relit result rather than a reusable rendering representation.

Relation to This Thesis. Taken together, these works clarify the design space in which this thesis operates. One option is to use diffusion priors to solve inverse and forward rendering jointly, as in DiffusionRenderer. A second is to move toward end-to-end relighting models that internalise decomposition, as in UniRelight. A third is to make intrinsic channels themselves the central interface for both analysis and synthesis, as in RGB↔X. The present thesis is closest to the third option, but with a stronger emphasis on explicit modularity and a clearer progression across problem settings.

An additional difference is methodological. In most of the recent related methods, relighting is still treated primarily as a probabilistic image-generation or video-generation problem: given a lighting condition and an image or latent representation, the model samples a plausible relit result from a learned conditional distribution. In this thesis, by contrast, relighting is modelled as a more deterministic task conditioned on explicit scene attributes. Once material, geometry, view direction, and illumination are specified, the desired outcome is not an arbitrary plausible sample but a constrained rendering result implied by those conditions. This difference explains why the thesis repeatedly favours explicit decomposition and rendering modules: they make illumination control behave more like a reproducible graphics operation than a purely stochastic appearance transformation.

More broadly, these recent works suggest a clear shift in emphasis within the literature. Rendering, decomposition, and synthesis are increasingly connected through editable intermediate representations with some physical meaning. The present thesis follows the same general direction, but studies it through a staged investigation of controllability across latent, 2D, and 3D generative models.

Chapter 3

Generative Fields for StyleGAN Based Image Synthesis Control

This chapter addresses **RQ1** by studying controllability inside a pre-trained StyleGAN2 generator. The practical task considered here is reference-guided face editing: given a generated identity image and a reference image supplying pose and expression, the output should preserve the identity of the generated face while matching the pose and expression of the reference.

The main goal of the chapter is twofold. First, we analyse which parts of the generator contribute to coarse and fine feature control and drawing inspiration from receptive fields in convolutional networks, we introduce the notion of *generative fields* as a quantitative way to describe the spatial scale influenced by a generator layer. Then we use that analysis to design a more explicit editing pipeline in StyleGAN2 style space. The resulting method is able to perform simultaneous generation and editing (SGE) during inference, without the need for an additional adversarial loss. We evaluate the method on a reference-guided face editing task and show that it can successfully transfer pose and expression from the reference image while preserving identity, achieving better performance than a comparable baseline.

3.1 Introduction

Generative adversarial networks (GANs) have become capable of synthesising highly realistic faces, but precise control over individual attributes remains difficult. This difficulty is especially visible in editing tasks where identity should remain stable while pose or expression changes. In such settings, the challenge is not only to produce a plausible image, but to decide where in the generator the control signal should act and how strongly it should influence the result.

Existing approaches to controllable GAN editing usually operate in one of three ways.

Some methods learn latent directions or trajectories that steer synthesis in a chosen semantic direction. Others use text or other abstract conditioning signals. A third line of work uses example images to transfer pose, expression, or style from a reference image. These approaches can work well in practice, but they often leave the mechanism of control implicit: the model is known to respond to a control signal, but the spatial scale and generator location of that control are not clearly quantified.

This chapter focuses on a direct reference-guided editing setting using facial landmarks and head pose. This form of control is more explicit than text-based guidance and is closer to existing graphics and content-production workflows. The key hypothesis is that the scale of the editable feature should match the scale of the generator layer receiving the control signal. To test this idea, we introduce a quantitative notion of generative field size and use it to analyse the relationship between generator layers, style-space modulation, and editing performance.

Based on this analysis, we propose an image editing pipeline that directly modulates the channel-wise style space of StyleGAN2 using a reference signal. The chapter therefore contributes both an analysis of controllability and a concrete editing method. Its main contributions are as follows:

1. We use the concept of generative fields, inspired by receptive fields of convolution neural network to explain the hierarchical feature control with different fineness levels for the input of each convolution layer within the generator and provide a quantitative experimental analysis.

2. We analyze the principle of style space and connect it with the function of generative fields, showing how it enables fine-grained feature control. Then we design a new image editing pipeline for the pre-trained StyleGAN2 generator by using the style space and demonstrate improved performance on the image editing task.

3. We further evaluate the sparse property of feature editing control in style space, revealing that style space can be used to stack multiple control signals with different fineness levels, in a way consistent with our generative fields analysis.

3.2 Related Work

3.2.1 Generative adversarial networks

Generative Adversarial Networks were proposed in 2014 [37], fast becoming one of the important generative models. They are inspired by game theory: two models in which one is generator in charge of generating the result sampling on a probabilistic distribution, the other one discriminator is a critic model in charge of examining the generated result through

comparing to the input example; the two models are competing with each other during the training process, making them simultaneously stronger. Radford et al. [123] proposed DCGANs importing the convolution operation into the GANs model, which decreases the computation complexity to synthesize high-resolution images. Karras et al. [71] proposed StyleGAN, a redesign of the generator's structure replacing the initial input with inputs at multiple levels of the generator network, introducing the style signal that is merged into the intermediate generating feature maps by using AdaIN [57] within each generator block. They claimed this design can separate content and style generation processes. The quality of StyleGAN was further improved by subsequent versions [72, 69] leading to better image quality and training efficiency. Our model leverages the separation of content and style generation in StyleGAN2 [69] and focuses on investigating fine-grained control of these. We use the pre-trained StyleGAN2 as the backbone and design the auxiliary networks for the control task, which adaptively modulate the style signal while maintaining the high quality of image synthesis.

3.2.2 Semantic feature learning and editing

GAN-based approaches can generate high-fidelity images, but controllable editing remains difficult because the relationship between latent codes, generator layers, and semantic attributes is only partially understood. In an early study, Zeiler et al. [177] showed that intermediate layers of convolutional networks respond to image features of increasing complexity. Bau et al. [6] extended this kind of analysis to GANs and showed that generator channels can be associated with interpretable image structures. These studies motivate the idea that controllability may depend on where and how a control signal is injected into the generator.

Prior editing methods include latent-direction methods [16, 125, 60], decomposition-based approaches such as EigenGAN [48], and reference-guided image editing methods such as IDDisen [116]. In the context of face editing, these methods are related to reenactment-style tasks because the desired output combines identity preservation with transfer of pose or expression from a reference. However, most of them do not explicitly quantify why a particular latent space or layer is suitable for controlling a given scale of facial change. Our method is closest to Nitzan et al. [116], but differs in two important ways: it operates in StyleGAN2 style space $\mathcal{S}$ rather than only in $\mathcal{W}$, and it uses the generative-field analysis to justify where control should be applied.

3.2.3 Receptive fields of convolution neural networks

Receptive fields in convolution networks measure the extent to which the input signal may affect the output [105]; this is inspired by the neuroscience notion from the human

visual system. Le et al. [84] propose the calculation method of receptive fields, which is a cumulative process from input image to output feature map, it can measure the receptive scale for output features. Receptive fields are important in designing the structure of convolution networks; for example in object detection models, the object scale should match the appropriate receptive field size for correct feature extraction [19]. State-of-the-art approaches such as SSD, YOLO, Faster R-CNN, etc., use anchor-based techniques for multi-scale object detection, the position where the anchor layer should be placed is a critical consideration for the model design due to the influence of receptive fields [34, 188, 187, 28]. For the generative model, Jaipuria et al. [61] analyzed the receptive field function of the GAN's discriminator and its effect on image generation, which is similar to normal convolution networks. In our work, we quantitatively evaluate the influence between inverted receptive fields and feature editing results, disclosing the high relevance between the injected control signal position and the fineness level control in synthesized results.

3.3 Method

3.3.1 Task Definition

The task studied in this chapter is *reference-guided facial editing* for a fixed pre-trained StyleGAN2 generator G. Let w_{id} denote a latent code sampled for the generator and let

$$I_{id} = G(w_{id})$$

be the generated identity image. Let I_{attr} be a reference image that provides the target pose and facial-expression information. The goal is to compute an edited output image I_{out} such that the identity comes from I_{id} while the controllable attributes come from I_{attr}.

More formally, the intended mapping in this chapter can be written as

$$I_{out} = G\big(w_{id}, c(I_{attr})\big),$$

where $c(I_{attr})$ denotes the control signal extracted from the reference image. In our case, this signal is defined by head pose and facial landmarks, and is injected into the generator through intermediate style representations rather than by modifying pixels directly. The desired behavior of the mapping is:

- the identity of I_{out} matches the identity of I_{id},
- the pose and facial expression of I_{out} match those of I_{attr},
- the output remains on the image manifold of the pre-trained generator.

This formulation differs from standard image-to-image translation because the source identity image is generated, not observed, and because the control is applied inside the generator rather than through pixel-space warping. It is also related to face reenactment, but in our case the main object of study is the controllability of StyleGAN itself: the method is designed to test how internal generator representations can be used for controlled editing, rather than to design a dedicated reenactment network operating purely in image space.

3.3.2 StyleGAN image generation

StyleGAN introduced a hierarchical generative process for generating human faces. Contrary to classical GAN approaches the generating noise is injected at multiple processing blocks separately. In this paper, we use the version proposed in StyleGAN2 [69], although the approach could be used similarly on the original StyleGAN.

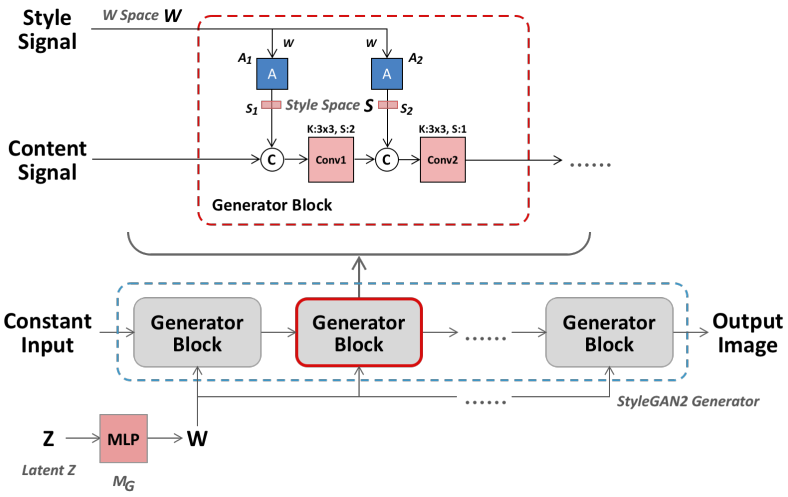

Figure 3.1: Image generation process in StyleGAN2. Bottom shows the whole generation pipeline including content process and style sampling process; top shows the detailed style modulation process.

The StyleGAN2 architecture, illustrated in Figure 3.1, is composed of several generator blocks [72]. In contrast to typical GANs, the *content signal* input to the first block is a constant vector only responsible for generating rough output; the main factor driving the variety and vividness of generated images is the *style signal*. Specifically, each generator block except the first consists of two affine transformations A_1, A_2 and two convolution layers `conv1`, `conv2` with same the kernel size 3×3 and different strides $(2, 1$ respectively); the first generator block only has one convolution layer with 3×3 kernel size and stride of 1; we omit the RGB convolution layer in each block since they are not relevant of controlling the main feature synthesis [162]. Each generator block upscales the input feature map to twice its size, the number of generator blocks N is therefore dependent on the desired

resolution of the synthesized image. We use the 256×256 resolution model composed of $N = 7$ generator blocks (13 convolution layers for style space [162]) in our experiments, we denote the convolution layers as `conv0` to `conv12`. For a pre-trained StyleGAN2 model, the image generation process is that the content signal is set from the initial learnt constant and gradually builds the feature map through each generator block; the style signal, a 512-dimensional vector z, is sampled randomly from initial latent space $\mathcal{Z}$ and then transformed by a dense model M_G to gain the latent vector w for better feature disentanglement [71], forming a new latent space $\mathcal{W}$. The latent vector w is replicated for the number of generator blocks to be the input of each one, providing the variety of the style signal, the overall latent vectors w form another latent space $\mathcal{W}+$ [162], we denote it as W_+.

Within each generator block, for each convolution layer `convi` ($i \in [1, 2]$), an affine transformation A_i is learnt that maps the input latent vector $\mathbf{w}$ to a style vectors $\mathbf{s}_i$ with dimensions equal to the input channel of `convi` [72]. Each style vector provides functional style information to modulate the visual feature synthesis[1] [162]. All style vectors form a new latent space $\mathcal{S}$, of dimension 4,928 for our architecture.[2] To simplify notation, we define the overall 4928-dimensional style vector $S = [S_d]_{1 \leq d \leq 4928}$ as the concatenation of the style vectors for all convolution layers and A as the complete mapping from W_+ to S such that $S = A(W_+)$.

3.3.3 Generative fields

Whether applied to classification or generation tasks, training convolution units implies learning spatial semantic information. The locality of convolution kernels implies that features at different layers encode patterns of different granularity. For generative models, early layers control global features whereas later layers control more local features (the converse is true for classification CNNs) [177, 6, 71, 162, 48]. In the case of perceptive CNNs, this is explained by the concept of receptive fields [175]. We extend this intuition to generative models through the notion of *generative field*, a generative counterpart of receptive-field size in discriminative CNNs and can be computed from kernel sizes and strides. Equation 3.1 gives the definition used in this chapter:

$$g_0 = \sum_{l=1}^{N-L} \left((k_{N-l+1} - 1) \prod_{i=N-l+1}^{N} s_i \right) + 1 \tag{3.1}$$

[1]The specific style modulation operation varies for different StyleGAN versions: StyleGAN1 uses AdaIN [57], whereas StyleGAN2 uses mod-demod operations [72]. We omit the noise signal modulation since it doesn't control the main factor of synthesized features.

[2]We use the definition of style space from [162] where the dimension is corresponding to all input channels of `conv1`, `conv2` from each generator block.

where k_l, s_i are the kernel and stride size from l-th, i-th layer of convolution calculation , g_0 is the generative field size of input feature map of $(L+1)$-th convolution layer (L $\in [0, 12]$), N is the total number of convolution layers (we use the start point of L from 0th for the consistency of style space).

As the example of feature generation in StyleGAN2, the position of convolution unit in the generator network determines various generative fields. Figure 3.2 depicts the generative field of chosen convolution layers for input feature map of sizes $8\times8, 32\times32, 128\times128$ in the StyleGAN2 generator. As illustrated in Section 3.3.2, all convolution layers can be indexed from `conv0` to `conv12` for 256×256 resolution model, we calculate generative fields of convolution layers whose indices are `conv2`, `conv6`, `conv10`, the 3^{rd}, 7^{th}, 11^{th} convolution layer of StyleGAN2 generator, giving generative field sizes of $251 \times 251, 59 \times 59, 11 \times 11$ respectively, which are the typical size corresponding to large, middle, small features in a model generating images of resolution 256×256. The content generation process of StyleGAN2 is a composition of synthesized features with generative fields sizes from small to large, combining all information in a coherent whole. The detailed generative field sizes for all convolution units are summarized in Appendix A.

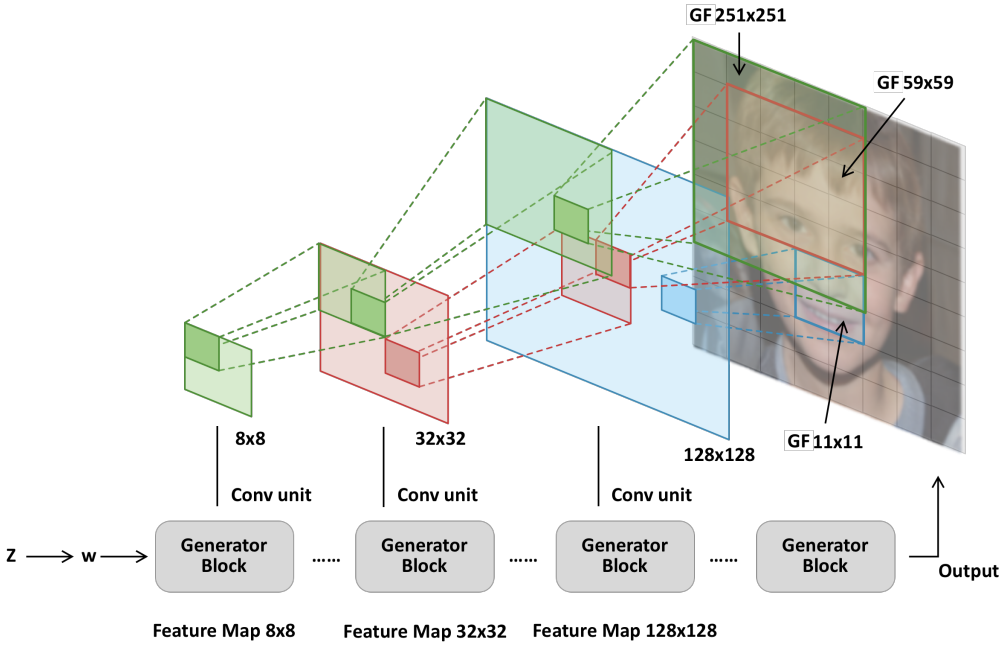

Figure 3.2: Generative fields produced by convolution units at different StyleGAN2 generator blocks. The leftmost unit feature map size is 8×8 and controls the largest generative field, of size 251×251; conversely, the rightmost unit feature map size is 128×128 and controls the smallest generative field, of size 11×11.

Specifically, a convolution layer with large generative field size (eg, `conv2`) should allow the control of global style features, such as head pose, because its generative field size covers the whole image. Similarly, convolution layers with smaller generative fields

(eg, `conv6`) should control more fine-grained style features such as facial expression. Finally, convolution layers with the smallest generative fields (eg, `conv10`) should have no structural impact, but refine the tone and texture of the generated image. Intuitively, this analysis is compatible with the style mixing experiments in the original StyleGAN paper [71]. On the other hand, style vectors as the input of `conv2`, `conv6`, `conv10` in style modulation process determine the editing information with various fineness level for final result, indicating that the style space $\mathcal{S}$ formed by all style vectors (Section 3.3.2) could be used to edit synthesized feature in different fineness level.

3.3.4 Feature control for StyleGAN2

Based on our analysis of generative fields (Section 3.3.3), we design a head feature editing pipeline working on style space $\mathcal{S}$ of a pre-trained StyleGAN2, adaptively utilizing channel-wise latent space with multiple generative fields for fine-grained control of human face synthesis. Given a reference image as the style input, head pose and facial expression features are extracted by a set of neural networks as the control signal, which is then injected into the style space of each convolution generator block, adjusting the style modulation during generating. Another image coupled with its sample vector provides identity information for the synthesis process. In order to encode facial expression and head pose features, we use a combination of facial landmarks and head pose Euler angles, as illustrated in Figure 3.3.

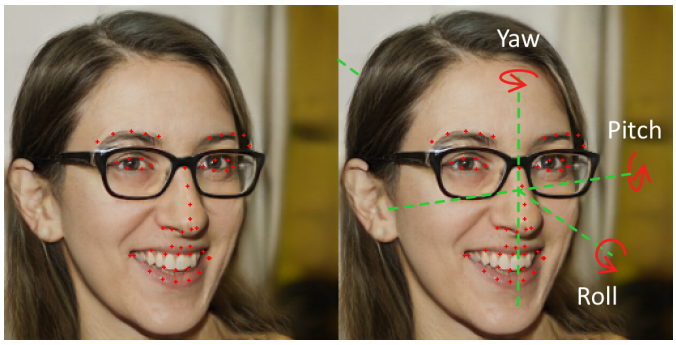

Figure 3.3: Facial landmarks (left) and head pose Euler angles (right).

The facial landmarks are defined by 68 feature points on the human face, used to encode the facial pose and expression information. Each feature point is denoted by 3-dimensional point coordinates. Due to the first 17 landmarks being dependent on face shape (and therefore identity) rather than expression, we only use 51 inner-landmarks from index 18 to 68 for better editing results, which are shown as red colour points in Figure 3.3. The right panel illustrates the definition of Euler angles: yaw, pitch and roll. The each angle is in the range $(-\frac{\pi}{2}, +\frac{\pi}{2})$. These angles are used to define the head pose, combined with facial landmarks to describe the overall human facial features.

Feature control architecture

Our image editing pipeline, shown in Figure 3.4, extends the approach proposed from [116] by 1) controlling explicitly for pose, and 2) modulating the control signal at each generator block. We modify the StyleGAN2 generator [72] to input the *control signal*, containing face pose and expression, into the style space.

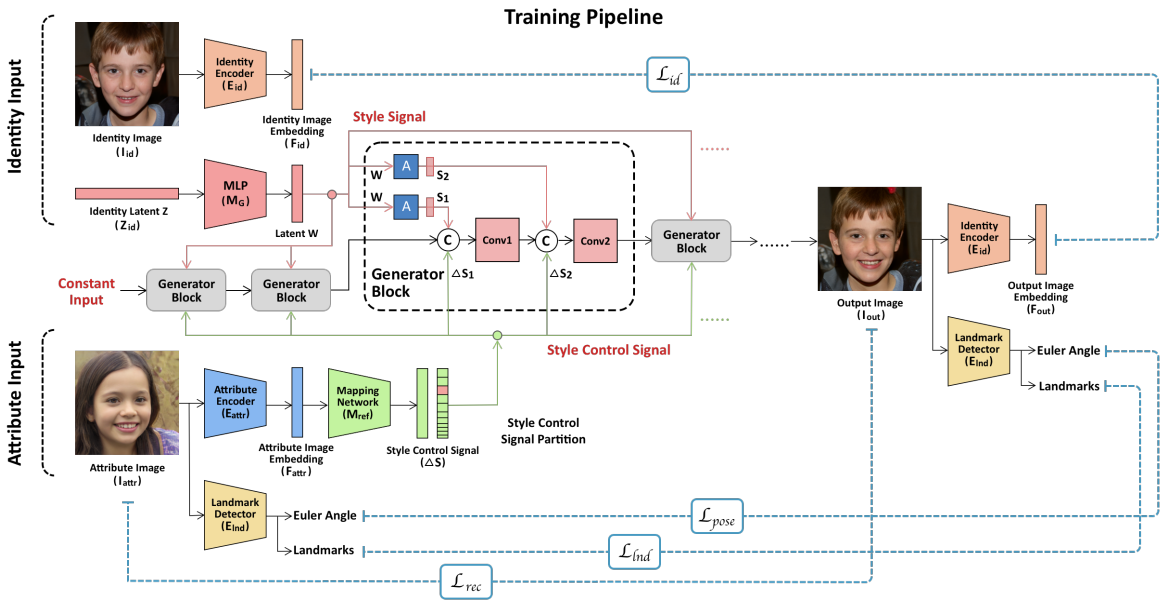

Figure 3.4: Image editing pipeline for StyleGAN2 using style space $\mathcal{S}$. Identity input including latent vector Z and corresponding generated image I_{id} for the facial generation with identical features. The attribute input is a reference image I_{attr} from which we extract facial features (expression, head pose) for controlling the image generation. All control signals work within each generator block, modulating the style signal samples in layer-wise style space $\mathcal{S}$.

The architecture is composed of 5 networks: The StyleGAN2 generator G, the identity encoder E_{id}, the attribute encoder E_{attr}, the reference mapping network M_{ref}, and the facial landmark detector E_{lnd}. In our approach, G, E_{id}, $E_{\text{attr}}$and E_{lnd} are pre-trained and only M_{ref} is trained from scratch. There are 2 types of input, 1) identity input consists of the StyleGAN-generated image $I_{\text{id}} = G(A(M_G(Z_{\text{id}})))$ that we want to edit and its latent sampling code Z_{id} for StyleGAN2; 2) the attribute input I_{attr}, which is a reference image providing the style information that the identity image should be changed to mimic. During training, the networks E_{id} and E_{attr} will encode the images I_{id} and I_{attr} into the feature representations F_{id} and F_{attr}, respectively:

$$F_{\text{id}} = E_{\text{id}}(I_{\text{id}}) \tag{3.2}$$

$$F_{\text{attr}} = E_{\text{attr}}(I_{\text{attr}}) \tag{3.3}$$

M_{ref} learns a transformation from the feature embedding of the attribute image F_{attr} to the style space $\mathcal{S}$, to produce a control signal that modulates the style signal I_{id}, editing the features by offsetting the channel-wise numerical value [162]. The StyleGAN2 generator G then synthesizes the edited image I_{out} through modulated style vectors:

$$\Delta S = M_{\text{ref}}(F_{\text{attr}}) \tag{3.4}$$

$$S_{id} = A(M_G(Z_{\text{id}})) \tag{3.5}$$

$$I_{\text{out}} = G(S_{id} + \Delta S) \tag{3.6}$$

where Z_{id} is used to recover the generation of identity image in StyleGAN2, S_{id} is the style signal gotten from Z_{id}, M_G and A denote the latent mapping network of StyleGAN2 and affine transformations of generator blocks. Referring to the style-signal definition in Section 3.3.2, we define $\Delta S = [\Delta s_d]_{1 \leq d \leq 4928}$ as the style control signal concatenating all style control vectors $\Delta \mathbf{s}$ across all generator blocks, which is a 4928-dimensional vector corresponding to the dimension of S_{id}.

In contrast, [116] only uses $\mathcal{W}$ space as the controlling vector space and therefore requires an additional adversarial learning process to regularize the initial sampling, whereas our method uses the style space that just provides the control signal rather than the whole style signal, dividing up the image synthesizing and editing completely, achieving the ability of simultaneously generating and editing (SGE) during inference.

Loss functions

The loss functions we used are modified from [116] to account for the differences in our architecture. Considering the head pose rotation is a movement in 3D space, we use a 3D landmark detector from [192] to extract 3D facial landmarks and Euler angles from facial images. Moreover, because Nitzan's approach predicts the new $\mathcal{W}$ space vector conditioned on the attribute image input, the new $\mathcal{W}$ vector can deviate from the original data manifold failing to produce a valid face for the following task. They resolve this by pre-training the mapping network with an adversarial loss [116]. One benefit our editing features in style space directly is that our approach preserves the content synthesis signal, and therefore does not require the addition of an adversarial loss.

Identity loss: $\mathcal{L}_{id}$, is used to ensure that the transformation performed by the network preserves the identity of the person in the input image. It is implemented as the L_1 distance between the identity embedding of the identity image F_{id}=$E_{\text{id}}(I_{\text{id}})$ and of the generated image F_{out}=$E_{\text{id}}(I_{\text{out}})$:

$$\mathcal{L}_{id} = \parallel E_{id}(I_{id}) - E_{id}(I_{out}) \parallel_1 \tag{3.7}$$

Attribute loss: $\mathcal{L}_{attr}$ is used to ensure that the generated image's pose and expression are

similar to the attribute image's. It is implemented as the combination of the L_2 distance between detected facial landmarks and estimated Euler angles α, β, γ between I_{attr} and I_{out}.

$$\begin{aligned} \mathcal{L}_{lnd} &= \| E_{lnd}(I_{attr}) - E_{lnd}(I_{out}) \|_2 \\ \mathcal{L}_{pose} &= \| E_{pose}(I_{attr}) - E_{pose}(I_{out}) \|_2 \\ \mathcal{L}_{attr} &= \mathcal{L}_{lnd} + \mathcal{L}_{pose}, \end{aligned} \tag{3.8}$$

where E_{lnd}, E_{pose} are all provided by pre-trained 3D landmark and pose detectors.

Reconstruction loss: We use the reconstruction loss $\mathcal{L}_{rec}$ from [116] to preserve pixel-level information if I_{id}, I_{attr} are same, which is a weighted sum of pixel-wise L_1 loss and MS-SSIM loss:

$$\begin{aligned} \mathcal{L}_{struc} &= \text{MS-SSIM}(I_{attr}, I_{out})) \\ \mathcal{L}_{mix} &= \alpha \mathcal{L}_{struc} + (1-\alpha) \| I_{attr} - I_{out} \|_1 \end{aligned} \tag{3.9}$$

where α is the mixing hyperparameter, for which we use 0.84 as suggested by [191].

The training process is divided into the cross stage and the reconstruction stage, where the former is training for pose and expression transfer and the latter is training for content consistency. During cross stage the images I_{id} and I_{attr} are different and the model is trained to generate the image capturing features from I_{attr}; during the reconstruction stage the images I_{id} and I_{attr} are the same and the model is trained to reconstruct the inputs, L_{mix} is only employed when I_{attr}= I_{id}:

$$\mathcal{L}_{rec} = \begin{cases} \mathcal{L}_{\text{mix}}, & I_{\text{attr}} = I_{\text{id}} \\ 0, & \text{otherwise} \end{cases} \tag{3.10}$$

Total loss The overall loss is the weighted sum of the above losses, 3 parameters λ_1, λ_2, λ_3 are used to control the factor of each one, influencing the identity preservation, attribute feature editing and general attribute preservation in generated results:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{id}} + \lambda_2 \mathcal{L}_{\text{attr}} + \lambda_3 \mathcal{L}_{\text{rec}} \tag{3.11}$$

3.3.5 Style space regularization

We found that training could become unstable after several epochs. We explain it through the derivation of data manifold in style space. The output image should capture the facial landmarks of I_{attr}, however, the loss produced from a bad-quality output image could be lower than a high-quality output image because E_{lnd} may also recognize a bad-quality human face and detect its landmarks, leading to the modulated value of style space

deviating from the data manifold.

To fix it, we propose a style regularization term which models the sampling of each style space channel as a Gaussian distribution (and for simplicity, assumed independent). Because the sampling value of the style space for the training dataset is known, we can get the mean μ_i and standard derivation σ_i of the training examples for each style channel s_i. We therefore can also compute the log-likelihood $L(S)$ for each style space channel s_i on every training batch, given by

$$L(S) = -\frac{1}{2\sigma_i^2}\sum_{i}^{|S|}(s_i - \mu_i)^2 \tag{3.12}$$

where μ_i, σ_i are mean and standard derivation of style code from the generated dataset, s_i is the style sampling of i-th channel of style space.

We can then maximise the log-likelihood to restrict the drift from the data manifold when searching style vectors. The style regularization term $-L(S)$ is added to the loss function during optimization. In our ablation experiments, we show this noticeably improves editing performance.

3.3.6 Implementation details

We use a pre-trained StyleGAN2 model with 256×256 resolution in all experiment tasks, the training pipeline is composed of 2 stages, cross stage and reconstruction stage, and a hyper-parameter controls the ratio of them in charge of the attribute learning and reconstruction respectively, we choose the ratio 3 as the suggestion of referred work [116]. A pre-trained landmark detection model 3DDFA is used as E_{lnd} [192] to regress 3D coordinates of 68 facial key points and 3 Euler angles, facilitating the feature editing and embedding experiment.

The Adam optimizer [78] is used for all experiments with parameters β_1=0.9, β_2=0.999 and the learning rate is dynamically changed from 5×10^{-6} to 1×10^{-7} for quick convergence. We empirically set the loss weights to $\lambda_1 = 1, \lambda_2 = 0.01, \lambda_3 = 0.02$ and the hyper parameter of $\mathcal{L}_{\text{rec}}$ to 0.84. For the calculation of $\mathcal{L}_{\text{lnd}}$, we take the suggestion from the referred work to use 51 inner-face landmarks [116] as they report the jawline landmarks could strongly result in the unreal face generation. The model is trained with batch-size 6 on a single NVIDIA GeForce RTX 3090 GPU and the training process is very efficient, taking only one day to converge.

3.4 Experiments

3.4.1 Image editing

We evaluate Chapter 3 from two perspectives. First, we test the practical editing task defined above: the output should preserve identity while matching the reference pose and expression. Second, we test whether the proposed generative-field analysis helps explain which parts of the generator are responsible for different editing scales.

The evaluation metrics therefore cover both *realism* and *editing accuracy*. To measure realism, we use Fréchet Inception Distance (FID) [51]. Lower FID indicates that the edited outputs remain closer to the real FFHQ distribution.

Editing accuracy is measured by three task-specific metrics. *Identity* is measured by cosine similarity between the identity embeddings of the source and edited images, so higher is better. *Expression* is measured by Euclidean distance between normalized facial landmarks of the attribute and edited images, so lower is better. *Pose* is measured by mean squared error between the Euler angles of the attribute and edited images, so lower is better. These metrics were chosen because they correspond directly to the intended behaviour of the task: preserve identity while transferring pose and expression.

Figure 3.5: Image editing result. Identity images are generated from StyleGAN2 randomly, attribute images are the real image set sampled from FFHQ256 dataset, identity images should capture pose and expression from attribute images.

Considering the diversity of StyleGAN2, we use the generated dataset from pre-trained StyleGAN2 on FFHQ256 dataset, consisting of 48,000 high-quality synthesized images as the training data for our image editing pipeline, from which 36,000 images are used as

identity images and 12,000 images as attribute images. This ensures that a wide range of attribute variation is covered by model generation capacity. We use real-world attribute images from the FFHQ256 dataset for evaluation to test the model performance and generalization. Figure 3.5 demonstrates high-quality image editing results: the generated image captures the pose and expression of attribute images while preserving the identity of identity images.

Comparison with previous methods

We compare our approach with LORD [32], FSGAN [115], and IDDISEN [116]. LORD is a landmark-guided face reenactment method, FSGAN is a face-swapping and reenactment framework designed to preserve identity during attribute transfer, and IDDISEN is a reference-guided disentanglement approach that explicitly separates identity and editable facial attributes. These baselines are appropriate because together they cover the main families of methods most closely related to our task: landmark- or attribute-driven reenactment, identity-preserving facial editing, and reference-guided disentangled control. Comparing against them therefore tests whether our StyleGAN-based style-space control offers an advantage over prior image-space or disentanglement-based approaches under the same identity-preservation objective. For each method, we evaluate 1,000 randomly sampled identity-reference pairs and compute the average realism and editing metrics. Table 3.1 shows that our method achieves the strongest identity preservation and pose transfer, while remaining competitive on realism and expression transfer.

Table 3.1: Quantitative evaluation with previous methods

Method	FID ↓	Identity ↑	Expression ↓	Pose ↓
LORD	23.08	0.20	0.085	13.34
FSGAN	8.90	0.35	**0.013**	6.73
IDDISEN	**4.28**	0.60	0.017	9.74
Ours	4.89	**0.82**	0.017	**1.67**

Ablation study

We conducted an ablation study to isolate the effect of the main design choices in our editing pipeline, using the same evaluation protocol as above. The goal is to test whether the gains come from style-space control itself, from the style regularization term, or from the explicit pose loss.

In Table 3.2, the default configuration follows the feature-concatenation strategy of Nitzan et al. [116]. The remaining rows then test our modifications. Removing feature concatenation improves editing performance but degrades realism, suggesting that direct

Table 3.2: Quantitative evaluation of image editing pipeline

Configuration	FID ↓	Identity ↑	Expression ↓	Pose ↓
default configuration	3.65	0.84	0.026	3.29
– feature concatenation	8.52	0.77	0.016	1.62
+ style regularization	4.48	0.83	0.022	2.33
+ Euler angle loss	4.89	0.82	0.017	1.67

style-space modulation is effective for control but still benefits from additional regularization. Adding style regularization improves stability, and adding an explicit Euler-angle loss gives the best overall balance between image quality and pose-expression transfer. Figure 3.6 shows the same trend qualitatively.

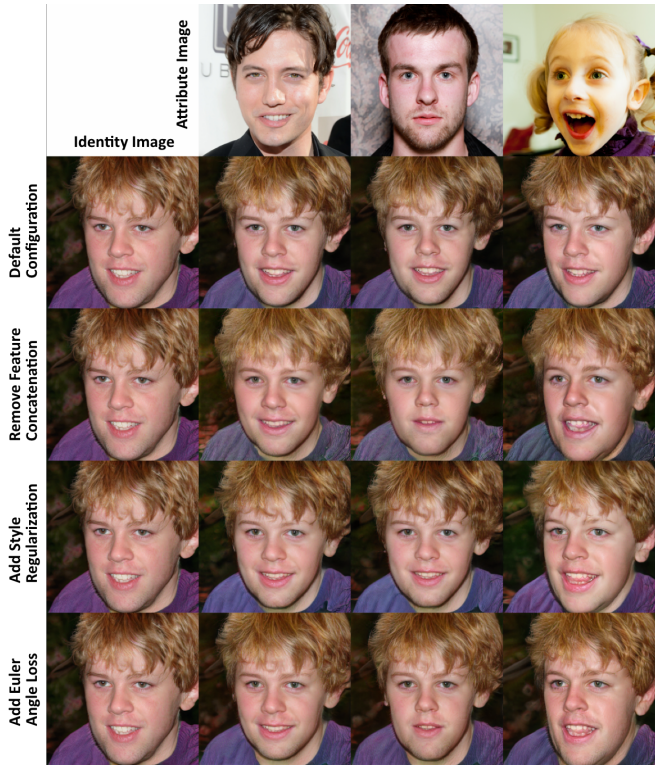

Figure 3.6: Comparison of image editing results with different experiment configurations in the ablation study. The second row results significantly improve the editing performance by removing the feature concatenation but loses some quality. The last row adds style regularization and Euler angle loss, getting the best overall performance with the balance between image quality and editing.

3.4.2 Generative fields

Sparsity of style space control

The first hypothesis tested in this section is that pose-expression editing in style space should be sparse: if identity is largely preserved, then only a small subset of style dimensions

should need to change substantially. We test this hypothesis with two experiments, one measuring the magnitude distribution of the control signal and the other measuring how consistently the same dimensions are reused across editing tasks.

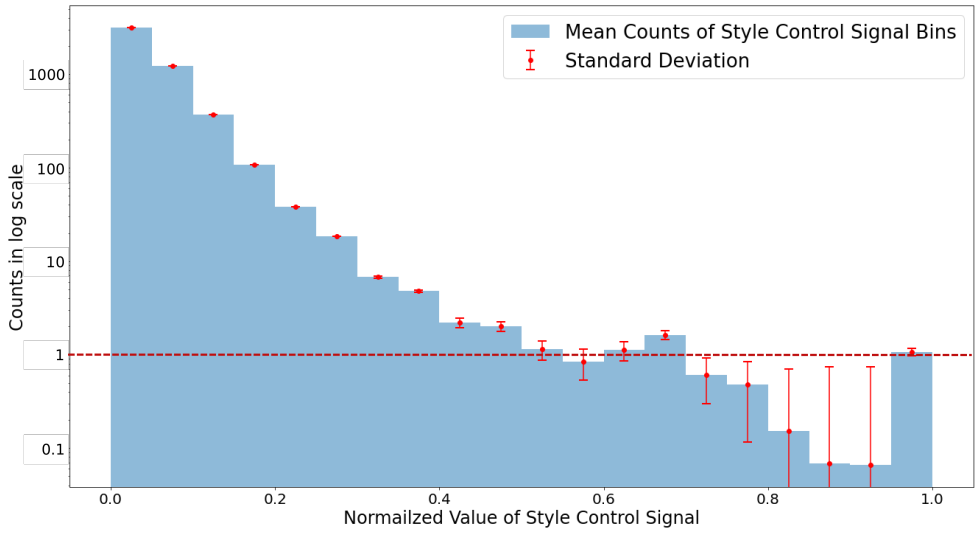

Figure 3.7: Histogram of mean style control signal ΔS. X-Axis is the normalized value of ΔS, Y-Axis is the mean counts of ΔS among all experiments locating on 20 data bins.

For the first experiment, we compute the histogram of absolute control values in ΔS over multiple editing trials. Because the scale of ΔS varies across examples, we normalize the absolute values before averaging histograms across tests. Figure 3.7 shows that only a small number of dimensions take large values, while most dimensions remain near zero. This supports the claim that style-space editing is sparse.

The second experiment asks whether the active dimensions are merely sparse or also reused consistently across different edits. We approximate the active set by taking the top 50 absolute entries of ΔS in each experiment and define a *reuse rate* to measure how often the same channel indices reappear across tests.

$$T = \{t \in \Delta S | \, \text{rank}_{\Delta S}(t) \leq 50\} \tag{3.13}$$

where t is the dimension value of ΔS, $\text{rank}_{\Delta S}$ is the absolute value rank in ΔS, rank 1 indicates the highest absolute value.

$$R_{reuse} = \frac{\text{card}(x)}{N}, \; x \in X \tag{3.14}$$

where x is the element of set X, N is the number of test, $card(x)$ is the cardinality of x in the set X.

As the style dimensions with the top 50 absolute values may vary in each experiment, we calculate the union set of corresponding dimension indices among all tests. Specifically, we conduct 10 feature editing tests and get 10 sets $T_1, T_2, T_3, ..., T_{10}$, then define the union

set of all dimensions T_u as:

$$T_u = T_1 \cup T_2 \cup T_3 ... \cup T_{10}. \tag{3.15}$$

Figure 3.8 shows that several channel indices are repeatedly activated across different editing trials. This suggests that pose and expression transfer in StyleGAN2 is not only sparse but also structured: a relatively small subset of channels contributes repeatedly to similar edits.

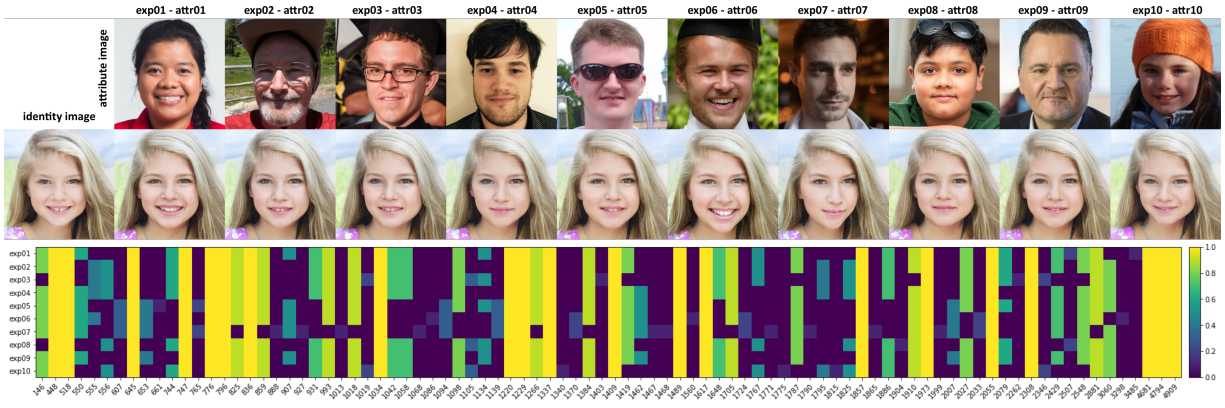

Figure 3.8: Statistics of channel reuse proportion for top 50 absolute control value of style space among all experiments, the yellow colour indicates the highest reuse rate R_{reuse}, the purple colour indicates the lowest reuse rate, the X-Axis is the channel index in the union set A_u of top 50 control channels among all experiments.

Generative fields evaluation

The second hypothesis of the chapter is that editing performance depends on matching the scale of the control signal to the scale of the target feature. If the active generator layers have generative fields that are too small, they should fail to control global attributes such as head pose. If they are too large, they should be less effective for fine-scale attributes such as expression. In our experiments, the average face width in FFHQ256 and the synthesized dataset is 141.68 pixels, measured using the 1^{st} and 17^{th} landmarks.

We design an experiment to test above hypothesis by disabling a set of style control vectors ΔS (Section 3.3.4) according to the generative field size from its input feature map and evaluating the editing performance. Specifically, we calculate the mean metrics result from 100 tests where identity images are randomly sampled from our synthesised dataset and attribute images are randomly sampled from FFHQ256 dataset. The StyleGAN2 generator with 256×256 resolution has 13 convolution layers from `conv0` to `conv12` (Section 3.3.3), we define the control units in each experiment configuration that keep the style control signal injection for these convolution units but set others to 0. For instance, the control units for **configuration 1** only keep `conv0` to `conv7` working for the feature editing, and set style control signal to 0 for other convolution units, which covers generative

fields size from 43 pixels to 506 pixels. Table 3.3 compares the relationship between model performance and the generative fields (GFs) of the functional control units.

Table 3.3: Quantitative evaluation of generative fields experiment

Index	Control Units	GFs	Identity ↑	Expression ↓	Pose ↓
config.1	conv0 - conv7	(43, 506)	0.75	0.016	0.99
config.2	conv0 - conv4	(123, 506)	0.74	0.019	1.04
config.3	conv0 - conv2	(251, 506)	0.76	0.028	0.67
config.4	conv3 - conv6	(59, 187)	0.73	0.017	1.21
config.5	conv6 - conv11	(7, 59)	0.69	0.033	0.84

These results support the hypothesis. Configuration 3, whose active layers have generative fields larger than the average face scale, performs best for pose control but loses expression accuracy. Configuration 5, whose active layers have much smaller generative fields, performs worst on identity and expression and produces visibly unstable edits. Together, these outcomes are consistent with the claim that coarse and fine facial edits require control to be injected at different spatial scales.

Figure 3.9: Small generative field works for the expression landmark editing, but the result gets the broken artifact due to limited influencing area, degrading the image quality.

3.5 Conclusion

This chapter showed that fine-grained editing in StyleGAN2 can be analysed and improved by studying where control is applied inside the generator. The proposed style-space editing method improves identity preservation and pose transfer relative to prior baselines, while the generative-field analysis provides a quantitative explanation for why different layers are better suited to different scales of editing. The experiments further indicate that the control signal is sparse and repeatedly activates a small subset of style dimensions across editing tasks.

At the same time, the chapter has clear limitations. The control variables are still defined in terms of facial landmarks and Euler angles rather than physics-based scene parameters, and the method remains tied to the representational capacity of a pre-trained StyleGAN2 face generator. Full 3D supervision is also absent, which limits control over viewpoint-dependent structure such as the back of the head. These limitations motivate the later chapters, which move from latent controllability toward explicit rendering-aware representations for illumination and viewpoint control.

Chapter 4

A Framework for Accurate Illumination Control

This chapter shifts the thesis from latent controllability to rendering-aware control. Chapter 3 showed that fine-grained editing can be improved by intervening on intermediate representations, but it still operated in a generator whose lighting was implicitly baked into the final image. Here we address **RQ2** 1.3 by studying the rendering stage directly and by asking whether illumination can be modelled as an explicit, editable process for real-world portrait images.

Precise illumination control requires modelling diffuse and specular light transport in a setting where material and geometry are not authored by an artist but estimated from photographs. This inverse setting introduces two practical difficulties that motivate the present chapter.

- **There is no material ground truth for real-world images.** The geometry and material attributes required for shading must be estimated from photographs rather than measured directly.
- **Traditional rendering is not well suited to estimated materials.** Classical rendering algorithms assume artist-authored geometry, materials, and calibrated lighting, whereas attributes recovered from real images are noisy and often violate those assumptions.

4.1 Introduction

Photo-realism is a key goal of computer graphics [113], and is essential for creating immersive multimedia experiences and imagery indistinguishable from the real world. Significant advances have been made in illumination representation, material appearance modelling, and high-precision geometric modelling [23, 24, 38, 154]. These techniques can

Figure 4.1: Example of rendering results from our physics-based neural deferred shader.

be understood as *forward approaches*: they explicitly model physical interactions between light and surfaces. However, they still rely on approximations of complex real-world physics, and these approximations can leave perceptible gaps between synthesized and real images.

In contrast, neural-network-based approaches learn aspects of the rendering process directly from real-world data; these can be viewed as *inverse approaches* or *data-driven approaches*, which reconstruct 3D information from 2D images. For example, Neural Radiance Fields [108] regress a scene radiance distribution with a neural network by estimating the color and density of sampled points, then use volume rendering to construct images from novel viewpoints. Gaussian Splatting [73] uses lightweight 3D Gaussians as the primitive representation, together with a fast approximate rasterization procedure to render them efficiently. Although these approaches can produce highly realistic images, they fit models that are specific to the content of individual scenes. They must be retrained from scratch for new scenes, and do not learn information that transfers or generalizes across scenes.

Another approach to photorealistic rendering is to use an auxiliary enhancement network to learn a mapping between the domain of traditionally rendered images and that of photorealistic images [77, 127]. As the initial rendering result lacks complete 3D scene information such as spatial distribution of illumination and 3D object positions, it is difficult for such post-hoc methods to learn a mapping to photorealistic images. In this chapter, we instead treat relighting as a task with explicit rendering inputs and outputs: estimated PBR textures, geometry cues, environment illumination, and camera parameters are given, and the goal is to predict the relit RGB image. We combine a deferred shading framework with a neural shading network, trained to directly regress pixel color from physics-based rendering (PBR) textures and light input, to achieve high-fidelity rendering results. Our contributions are as follows:

1. We propose a *physics-based neural deferred shading pipeline* that renders scenes with PBR textures (albedo, roughness, specular) and illumination (HDRI light map) into photorealistic images (Figure 4.1).

2. We develop a *neural shadow estimator* to efficiently approximate realistic shadows, improving the final rendering result.

3. We propose a *new FFHQ256-PBR dataset* containing RGB images, material textures (albedo, roughness, specular), depth, screen-space ambient occlusion, and HDRI environment maps.

4. We conduct comprehensive experiments to compare our approach with existing shading models on multiple datasets, demonstrating its superior performance.

4.2 Related Work

Photo-realistic Novel View Synthesis. Novel view synthesis is an application of the 3D reconstruction task, and recent learning-based techniques can reconstruct photorealistic 3D scenes from posed images and synthesize high-quality novel views from the recovered 3D information. Specifically, a series of works on Neural Radiance Fields [39, 108, 144, 174] represent a single 3D scene using an implicit radiance-field function: they sample points along camera rays and regress color and density to reconstruct each view. Other works on Gaussian splatting [73, 144, 171, 176] instead draw inspiration from point-based rendering and use a splatting procedure to render pixel colors. They use lightweight 3D Gaussians as the primitive representation and achieve real-time novel-view synthesis without sacrificing quality. Both families of methods require overfitting to specific scenes and are difficult to generalize because they compose geometry, material, and illumination together. In contrast, our work is based on the standard deferred rendering pipeline and learns a shading function that can render arbitrary scenes while preserving photorealistic appearance.

Photorealism Enhancement. Other works reduce the photorealism gap between traditional rendering output and real-world imagery by designing an enhancement model. Richter et al. [127] capture all shader inputs of the video game Grand Theft Auto 5 (GTA5) and use adversarial learning to increase the vividness of rendered results by discriminating them against matched real-world photos. Kim et al. [77] use a neural network to improve the output of a physics-based Cook-Torrance renderer; because paired inputs and targets are available, their method does not require adversarial learning. However, training a model to improve a forward-rendered image still restricts the learner to whatever information survives the initial rendering pass. Important variables such as view direction, per-pixel material parameters, and directional illumination are not preserved in sufficient detail for

robust relighting. In contrast, we propose a more direct shading formulation, in which a neural regressor models the shading process itself by predicting the RGB color of each rasterized shading point from informative PBR textures and illumination inputs, leading to better physical interpretability and stronger relighting behaviour.

Neural Deferred Rendering. In classical rendering, there are two common ways to shade a visible point: forward shading and deferred shading. Forward shading renders each object in turn and lights it according to all light sources in the scene. Deferred shading instead postpones expensive lighting computations to a later stage and performs them only for visible screen points, reducing computational cost [23]. Neural rendering is more commonly built on forward-shading-style formulations, albeit with scene representations learned from sample imagery [10, 40, 150, 169, 178]. However, a deferred pipeline can also be used in neural rendering, which is advantageous because it avoids the non-differentiable rasterization process in classical rendering and allows classical shading formulas to be replaced with neural models learned from data. Neural deferred rendering was first proposed by Thies et al. [151], who integrated the conventional real-time rendering pipeline with a learnable neural texture and a deferred neural renderer. Worchel et al. [159] later extended the deferred neural renderer to 3D reconstruction by using a differentiable rasterizer in the classical deferred pipeline to recover a triangle mesh from multi-view input. However, existing works on neural deferred shading do not model light-surface interaction explicitly and are not physics-based, making them unsuitable for generalization to new illumination conditions. Our work redesigns the coordinate-based neural deferred shader [159] by incorporating inbound light and bidirectional angular information so that the shading result is regressed from physically meaningful inputs.

4.3 Method

4.3.1 Task Definition

We formulate the chapter task as *real-world portrait relighting with estimated physics-based scene attributes*. The input consists of a G-buffer extracted from a single portrait image, including albedo, roughness, specular reflectance, normal, depth, and screen-space ambient occlusion, together with an HDRI environment map and camera field of view. The output is a relit RGB image under the specified illumination. This differs from scene-specific neural rendering methods such as NeRF or scene-fitted neural deferred renderers, which overfit to a single scene, and image-enhancement pipelines that only refine an already rendered image. Our task instead asks whether a scene-agnostic shader can learn the mapping from estimated material and illumination cues to photorealistic appearance for many real-world faces.

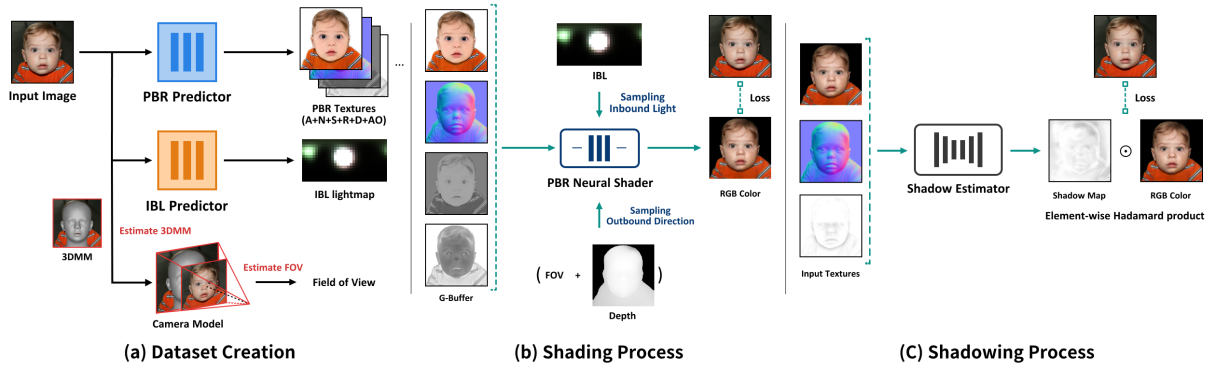

Figure 4.2: The overall pipeline of our physics-based neural deferred shading. In data preprocessing (a), the input image is processed to estimate PBR textures (A: albedo, N: normal, S: specular, R: roughness, D: depth, AO: ambient occlusion), an IBL light map, and the field of view via pre-trained models. The estimated data are then used to train the physics-based neural shader (b). Subsequently, a shadow estimator (c) predicts the shadow map applied to the final shading result.

Image rendering aims to model how light interacts with surfaces [77], as described by the rendering equation:

$$L_o(\mathbf{v}) = \int_{\Omega} F(\mathbf{v}, \mathbf{l}) L_i(\mathbf{l}) \langle \mathbf{n} \cdot \mathbf{l} \rangle d\mathbf{l} \tag{4.1}$$

where $L_o(\mathbf{v})$ is the outbound radiance leaving in direction $\mathbf{v}$; it is the integral of the incident light $L_i(\mathbf{l})$ from every possible direction $\mathbf{l}$ across the hemisphere Ω, centered around the surface normal $\mathbf{n}$. $F(\mathbf{v}, \mathbf{l})$ is the Bidirectional Reflectance Distribution Function (BRDF) describing how the surface reflects light.

In practice, approximations of this equation are used, limiting rendering fidelity. We introduce *physics-based neural deferred shading* in this work, a framework for photorealistic portrait rendering that replaces hand-coded approximations with a learned regressor conditioned on geometry, material, and light inputs. An overview of our pipeline is presented in Figure 4.2.

4.3.2 Physics-based Neural Deferred Shading

To approximate the rendering equation more precisely, our work redesigns previous neural deferred shaders [159] to incorporate PBR inputs, namely material textures, normal map, depth map, and HDRI light map. Because paired samples of PBR inputs and real-world target images are not publicly available, we first create a dataset for training and evaluation, then train the neural deferred shader and shadow estimator by minimizing the difference between rendered and target images. During inference, the model renders an arbitrary G-buffer containing estimated material and geometry information under a specified HDRI light map, producing a photorealistic image.

Dataset creation. We use the recent material and light predictor [77] to construct the

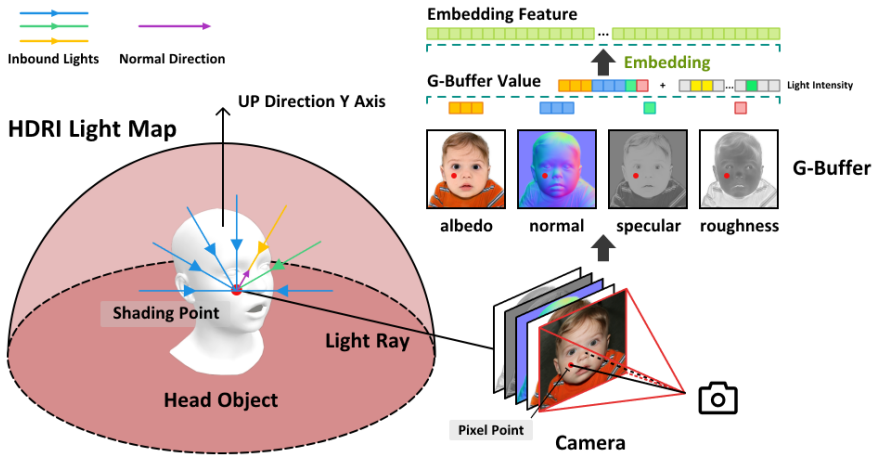

Figure 4.3: Shading process for each point. Each visible point collects its local PBR attributes to form a G-buffer entry, and sampled incident light from the HDRI map provides the directional lighting features used by the neural shader.

supervision data from images. The preprocessing pipeline has three stages, as summarized in Figure 4.2. First, we estimate physics-based rendering (PBR) material textures, normal map, depth map, screen-space ambient occlusion (SSAO), and an image-based lighting (IBL) environment map. The PBR textures consist of albedo, specular reflectance, and roughness, which are combined with the normal and depth maps to form the G-buffer used by the shader (see Section 4.2). Second, we estimate the field of view (FOV), which is required to recover the outbound direction of each pixel. To do so, we use a 3D Morphable Model (3DMM) to register input images and locate 3D landmarks on faces [29]; combining these landmarks with their 2D projections and the recovered depth map allows us to estimate the camera FOV. Third, we package the estimated attributes with the original RGB image for training and evaluation.

Using this process, we extend the human-face dataset FFHQ [70] into a new dataset called FFHQ256-PBR, consisting of 69,990 facial images with estimated PBR material textures, HDRI illumination, and camera field of view (See Figure 4.4). We manually remove obvious failure cases, such as background-removal errors that erase parts of the head or implausible material estimates around hair boundaries. This filtering step is important because it defines the usable supervision regime of the chapter: the dataset is not treated as measured ground truth, but as a large-scale estimated supervision set for studying whether a learned shader can compensate for the mismatch between inverse-rendered attributes and real images. In this sense, the dataset contribution of the chapter is not only the final collection itself, but also the preprocessing and curation procedure that makes large-scale portrait relighting experiments possible.

Our main experiments focus on human faces because they are central to many applications, but the rendering pipeline is not restricted to that domain. We therefore collect a

Figure 4.4: Visualization of the FFHQ256-PBR dataset. Each row shows the original RGB image, estimated albedo, normal, roughness, specular, depth.

second dataset based on a 1,000-scene subset of the synthetic BlenderVault dataset [99]. For this dataset, we estimate the relevant parameters using the inverse-rendering approach proposed in [99]. The two datasets play complementary roles: FFHQ256-PBR tests whether the shader can handle noisy real-world estimates, while BlenderVault provides a cleaner synthetic reference domain. Rather than relying on synthetic augmentation of FFHQ itself, we broaden the evaluation domain by pairing the real-world portrait dataset with this synthetic benchmark.

Shading process. We train the neural deferred shader to estimate the RGB color for each pixel, after which it can shade an arbitrary surface point during inference. Specifically, the shader processes each pixel of the G-buffer separately (Figure 4.3). Texture and normal information is represented in screen space and taken from the corresponding pixel, while illumination information is sampled from rays on the hemisphere around the 3D point found by casting a ray from the pixel into the scene.

The neural shader's input is inspired by the classical rendering equation (Equation 4.1), and consists of three material features corresponding to F term: albedo $\mathbf{a} \in [0,1]^3$, specular $\mathbf{s} \in [0,1]^1$ and roughness $\mathbf{r} \in [0,1]^1$; three geometric terms (as unit vectors): normal $\mathbf{n} \in [-1,1]^3$, inbound light direction $\mathbf{l} \in \mathbf{R}^3$ and outbound light direction $\mathbf{v} \in \mathbf{R}^3$; a single light feature: inbound light intensity $L_i \in (\mathbf{R}^+)^3$. The per-pixel values of $\mathbf{a}, \mathbf{n}, \mathbf{s}, \mathbf{r}$ form a G-buffer entry that is concatenated with the n sampled inbound light rays $\{L_i(\mathbf{l}_1), L_i(\mathbf{l}_2), ..., L_i(\mathbf{l}_n)\}$ to form the input of a trainable neural shader with parameter θ. Accordingly, the neural shading function is:

$$\int_{\Omega} f_\theta(\mathbf{a}, \mathbf{n}, \mathbf{s}, \mathbf{r}, \mathbf{v}, L_i(\mathbf{l})\langle \mathbf{n} \cdot \mathbf{l} \rangle) d\mathbf{l} \in [0,1]^3, \mathbf{l} \in \Omega^+ \tag{4.2}$$

where $L_i(\mathbf{l})\langle \mathbf{n} \cdot \mathbf{l} \rangle$ is the inbound light from the upper hemisphere Ω^+, analogous to Equation 4.1. The neural shader first regresses the outbound contribution for each sampled incident-light direction and then averages these contributions across the sampled rays to obtain the final result. In this way, the network learns to approximate the shading integral directly from reconstruction supervision.

Architecturally, the neural shader is a lightweight two-stage multilayer perceptron. After positional encoding [108, 146], a diffuse network with two fully connected layers predicts a view-independent diffuse feature. This feature is then concatenated with the outbound direction $\mathbf{v}$ and half-vector $\mathbf{h}$ and passed to a second two-layer specular network that produces the final RGB contribution for each sampled light direction. We separate the diffuse and specular branches because diffuse reflection is isotropic while specular reflection is explicitly view dependent; this design gives a clearer correspondence to the rendering equation than a single black-box regressor.

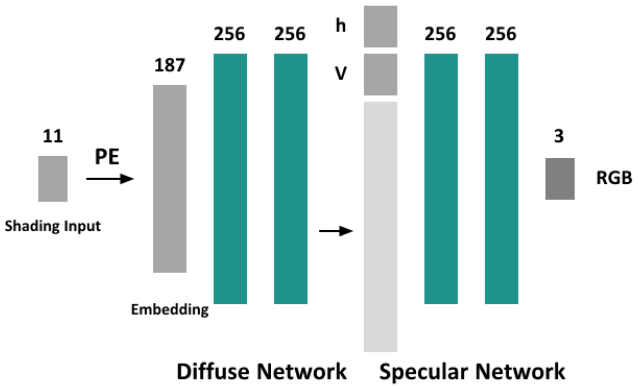

Figure 4.5: Architecture of physics based neural deferred shader. The PBR input is transformed through the positional encoding [146] and fed to the diffuse network. The resulting feature vector is then concatenated with both the outbound direction $\mathbf{v}$ and half direction $\mathbf{h}$ and fed as input to specular network, producing a RGB color value.

Specifically, it consists of two networks (see Figure 4.3): a diffuse network and a specular network. Both networks are multi-layer perceptrons (MLPs) consisting of 2 fully connected layers, using ReLU activation function. As MLPs are ill-suited to learn functions with high spatial frequency [59, 146], we first embed the G-buffer and light intensity into a high-dimensional Fourier feature via positional embedding [108, 159]. The diffuse network then processes the embedding features to produce the diffuse features. Note that since diffuse reflection is defined as isotropic, the view angle is not an input of this network. The diffuse features are then concatenated with both the outbound direction $\mathbf{v}$ and the half direction $\mathbf{h}$ and fed to the specular network to gain view-dependent specular reflection, and produce an RGB color for the pixel.

Shadowing process. The architecture proposed above does not account for how inbound light rays may be affected by self-occlusions—for example, the top of the head may shelter

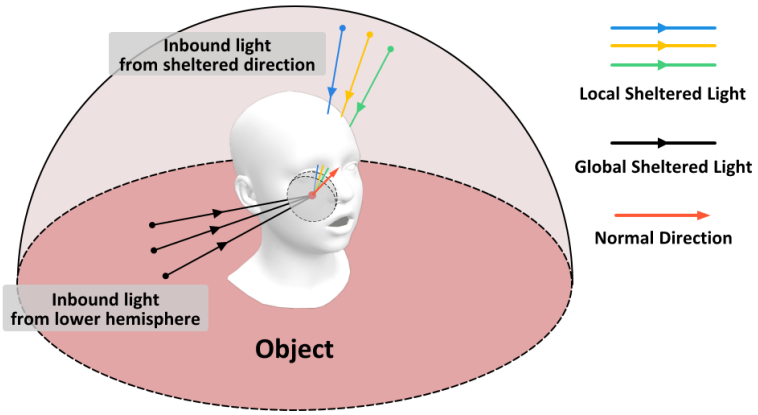

Figure 4.6: Localized shadowing model: The light rays denoted by black arrows come from the lower hemisphere and would be filtered by the cosine term in the rendering equation, whereas the light rays denoted by colorful arrows are occluded by local geometry resulting in localized shadowing.

light rays incident to a point below the eye, casting a shadow. Figure 4.6 illustrates this effect: light rays coming from the top side of the environment map are occluded by the top of the head. In practice, shadowing varies with differences in environment lighting: if the environment light map is even and smooth, screen space ambient occlusion (SSAO) can approximate the shadow effect well; conversely, if the environment light is highly anisotropic then the resulting hard shadows must be calculated by costly tracing of individual light paths. To calculate the shadow effect with a unified approach suitable for diverse light maps, we design a neural shadow estimator that learns to improve the shadowing result from our neural deferred shader. Specifically, a UNet-like neural network f_η is designed to estimate the shadow map from the input of an unshadowed image $\hat{I}_{unshadow}$ and corresponding geometry information (normal map I_{normal} and SSAO map I_{ssao}). The final shading result is the element-wise (Hadamard) product of the neural shader's output and the estimated shadow map. The shadowing process is thus described by

$$\begin{aligned} \hat{I}_{shadow} &= f_\eta(\hat{I}_{unshadow}, I_{normal}, I_{ssao}) \\ \hat{I}_{rgb} &= \hat{I}_{unshadow} \odot \hat{I}_{shadow} \end{aligned} \tag{4.3}$$

where $\hat{I}_{\text{shadow}}$ is the pixel-wise shadow map, η is parameters of the neural shadow estimator, $\hat{I}_{rgb}$ is the shadowed image.

4.3.3 Loss Function

The proposed model is trained by solving the following minimisation for the neural network parameters θ:

$$\underset{\theta}{\text{argmin}}\, L_{\text{appearance}}(\mathbf{a}, \mathbf{n}, \mathbf{s}, \mathbf{r}, \mathbf{v}, L_i) \tag{4.4}$$

where $L_{\text{appearance}}$ compares the rendered pixel to the ground truth pixel at the corresponding position. $\mathbf{a}, \mathbf{n}, \mathbf{s}, \mathbf{r}, \mathbf{v}, L_i$ are albedo, normal, specular, roughness, outbound direction, inbound light respectively.

More specifically, we train our model in two stages, with $L_{\text{appearance}}$ defined differently in each phase. In phase one, the model does not consider shadowing, so $L^{(1)}_{\text{appearance}}$ is the shading loss defined by Equation 4.5, which is an L_1 loss calculating the average distance between prediction and ground truth for all foreground pixels, N_S is the number of sampled pixels, and C is the RGB value of pixel.

$$L^{(1)}_{\text{appearance}} = \frac{1}{N_S} \sum_{i=1}^{N_S} \|C_i - \hat{C}_i\|_1 \tag{4.5}$$

We choose L_1 supervision because the target is direct appearance regression from noisy estimated attributes rather than adversarial image synthesis. In this setting, L_1 provides a stable objective that is less sensitive to outliers than L_2 and empirically preserves shading transitions better than a purely perceptual loss in our experiments.

In phase two, the neural deferred shader is frozen to produce an unshadowed image, which is fed to the shadow estimator through concatenating with the normal map (see Section 4.3.2), yielding the shadowed image $\hat{I}_{rgb}$. An L1 loss $L^{(2)}_{\text{appearance}}$ (Equation 4.6) calculates the residual between the shadowed image $\hat{I}_{rgb}$ and ground truth image I_{rgb} for all pixels.

$$L^{(2)}_{\text{appearance}} = \|I_{rgb} - \hat{I}_{rgb}\|_1 \tag{4.6}$$

4.3.4 Implementation Details

We train each stage for 40 epochs, which balances efficiency and performance. We use the Adam optimizer [79], with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a learning rate of 5×10^{-5}. We randomly sample 8,192 foreground pixels of each image to form each training batch; this provides enough diversity to avoid over-fitting as the neural deferred shading works on a per-pixel basis and is agnostic to the overall scene structure. We uniformly sample 128 inbound light rays from the HDRI light map as input to the shader in each training step. Training converges within one day on a single NVIDIA GeForce RTX 3090 GPU. During inference, the physics-based neural deferred shader can render an image in 1 second per frame.

4.4 Experiments

We conduct comprehensive experiments to evaluate the performance of our neural deferred shading approach, comparing it qualitatively and quantitatively against three baselines. The baseline models include two classical shading models, the empirical Blinn-Phong model and the physics-based GGX model, together with a recent learning-based neural deferred shader [159].

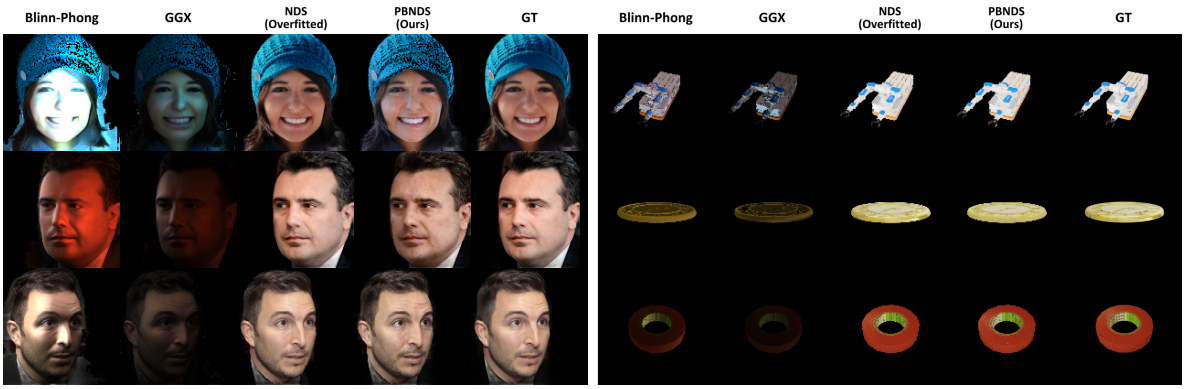

Figure 4.7: Quality comparison between different shading models. **Blinn-Phong**: Blinn-Phong shading model; **GGX**: Trowbridge-Reitz GGX model; **NDS**: neural deferred shader; **PBNDS**: physics-based neural deferred shader; **GT**: ground truth image. Our model reconstructs the scene while preserving realistic light-surface interaction and stronger generalization than the classical baselines.

4.4.1 Qualitative Evaluation

We visually compare shading results produced by different models from similar inputs. We evaluate examples from two datasets: a held-out test set from FFHQ256-PBR for facial data and a subset of BlenderVault [99] for other object types. Figure 4.7 illustrates the results for both datasets, comparing our neural shader with the classical Blinn-Phong [9] and GGX [154] models. Our physics-based neural deferred shader (PBNDS) reconstructs the image photorealistically from the estimated materials and illumination, and the overall hue and light reflection are more realistic than those of the classical models. In contrast, Blinn-Phong is an empirical local shading model and therefore cannot reproduce the richer reflection behaviour visible in the targets. The GGX model is based on microfacet theory, but it still assumes lighting and material inputs that are calibrated for forward rendering, making it less suitable for our estimated-attribute setting. We also compare our model to the state-of-the-art neural deferred shader (NDS) [159], and show that our approach can achieve similar quality. This is notable because our model is scene agnostic, whereas NDS overfits a separate neural renderer to each scene by taking scene-dependent point coordinates as input.

We next show how our model enables image relighting by replacing the HDRI map

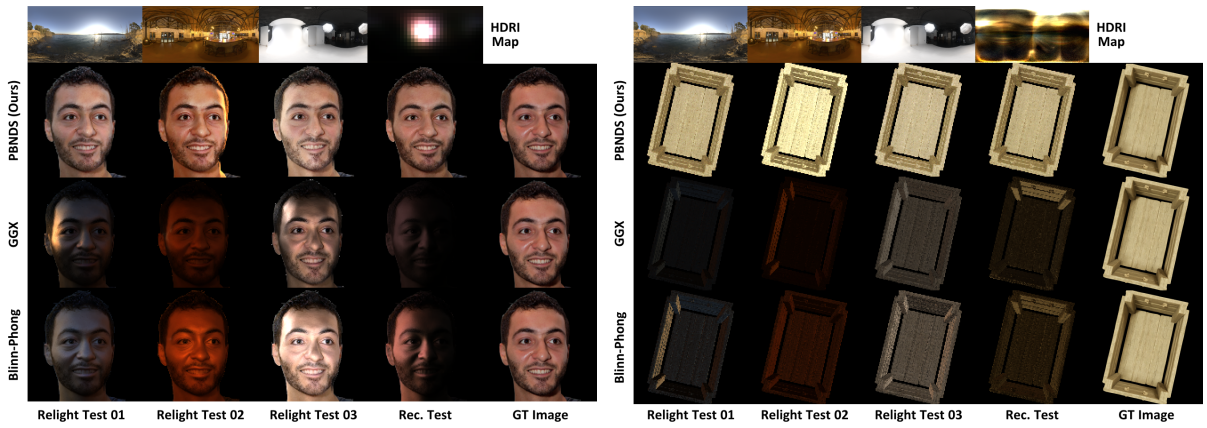

Figure 4.8: Rendering results with different HDRI maps. The first row shows HDRI maps used to relight the original head; the other rows show the resulting rendered images after relighting from different shading methods. The results show our neural shader allows the environment map to realistically influence the shading of the head.

used at rendering time. Specifically, we use the model trained in the previous section for both datasets but swap in new HDRI maps captured in real-world scenes. We compare the relighting results to those of the classical Blinn-Phong and GGX methods in Figure 4.8. Our model more faithfully recovers the light-surface interaction in this inverse setting, whereas the classical models respond poorly to the estimated material and illumination inputs. Compared with reconstructing the scene under its estimated illumination, the relighting test is harder because the model must adapt to real-world illumination conditions that may lie outside the distribution seen during training.

4.4.2 Quantitative Evaluation

We also compare the rendered images to the ground truth quantitatively. We conduct 500 tests in each experiment using data from the test split of FFHQ256-PBR and from BlenderVault [99]. For the shading experiment, we report Learned Perceptual Image Patch Similarity (LPIPS) [185] and Fréchet Inception Distance (FID) [50] to measure perceptual realism, where lower values are better, together with Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) to measure reconstruction accuracy, where higher values are better. For the relighting experiment, paired ground-truth targets are not available under each novel illumination, so we report only FID.

As shown in Table 4.1, our model substantially outperforms the classical shading models in both the shading and relighting experiments. This suggests that the learned shader adapts better to the mixed data domains used here, especially when rendering real-world scenes whose materials and illumination are estimated rather than calibrated for classical forward rendering. Our model also outperforms the overfitted neural deferred shader [159]

Table 4.1: Quantitative evaluation for shading/relighting experiments

		MSE ↓		PSNR ↑		SSIM ↑		LPIPS ↓		FID ↓	
		FP	BV	FP	BV	FP	BV	FP	BV	FP	BV
SD	Blinn-Phong	0.078	0.044	11.90	14.94	0.623	0.784	0.181	0.080	0.244	0.239
	GGX	0.127	0.055	9.11	13.35	0.489	0.753	0.273	0.114	0.368	0.227
	NDS (Overfitted)	**0.005**	**0.001**	**29.68**	**38.53**	0.885	**0.988**	0.062	0.027	0.179	**0.037**
	PBNDS (Ours)	0.007	0.002	24.61	29.47	**0.919**	0.936	**0.032**	**0.025**	**0.056**	0.113
RE	Blinn-Phong	-	-	-	-	-	-	-	-	0.331	0.163
	GGX	-	-	-	-	-	-	-	-	0.556	0.162
	PBNDS (Ours)	-	-	-	-	-	-	-	-	**0.090**	**0.117**

SD: Shading experiment; **RE:** Relighting experiment; **FP:** FFHQPBR dataset; **BV:** BlenderVault dataset

Table 4.2: Comparison of training with different model components

	MSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
Default Config.	0.0046	24.4107	0.9167	0.0337	0.0607
+ Perc. Loss	0.0052	24.0763	**0.9298**	0.0273	0.0441
+ SDO (w/o AO)	0.0041	24.6948	0.9194	0.0319	0.0559
+ SDO (AO)	**0.0039**	**25.0053**	0.9247	**0.0262**	**0.0240**

Table 4.3: Comparison of different training batch size

	PSNR ↑	GPU Mem.	FLOPs(G)
2048	21.74	2.98	0.64
4096	22.81	5.96	1.28
8192	24.83	11.91	2.56
Full Res.	25.81	23.82	5.12

in SSIM, LPIPS, and FID on the FFHQ256-PBR dataset, while remaining competitive in the more pixel-sensitive MSE and PSNR metrics across datasets. We attribute this behaviour to the fact that direct scene-specific overfitting can introduce high-frequency noise, especially on real-world images, whereas our global shadowing stage provides a more stable image-level constraint.

4.4.3 Ablation study

We conduct an ablation study to evaluate the importance of the different components of the neural shading pipeline, following the same experimental protocol as in Section 4.4.2. The default configuration of the neural deferred shader excludes the shadow estimator and uses only the reconstruction loss from Section 4.3.3, together with ReLU activations and positional encoding [108]. We then add one component at a time: a perceptual loss in the second test, our shadowing process without ambient occlusion (AO) in the third, and the full shadowing process with AO in the fourth. Table 4.2 shows that the perceptual loss improves perceptual metrics (SSIM, LPIPS, FID) but degrades the reconstruction metrics (MSE, PSNR). The shadowing process without AO balances these two aspects more effectively, and adding AO further improves most metrics while causing only a slight drop in SSIM. We therefore use the full shadowing process with AO as the final design. We also compare different training batch sizes and find that 8,192 pixels provide the best balance between model quality and training efficiency (Table 4.3).

4.5 Conclusion

This chapter showed that illumination control can be moved out of an implicit image generator and formulated as a separate rendering problem. By learning a scene-agnostic physics-based neural deferred shader from estimated PBR attributes, we obtained realistic portrait rendering and relighting results on both FFHQ256-PBR and BlenderVault, while remaining more general than methods that fit a separate neural renderer per scene. The chapter also introduced FFHQ256-PBR as a practical supervision source for studying relighting on real-world portrait data.

The chapter nevertheless has clear limitations. The supervision is derived from estimated rather than measured materials and lighting, so the model inevitably inherits errors from the inverse-rendering stage. In addition, the current formulation remains image-plane based, which means that viewpoint changes and self-occlusion are only approximated through estimated geometry cues rather than solved in a fully 3D representation. These limitations motivate Chapter 5, where the learned shader is coupled to a generative model so that the intermediate representation itself becomes editable, and Chapter 6, where viewpoint control is addressed in an explicit 3D setting.

Chapter 5

Decomposing Diffusion Models for Physics-Based Image Synthesis and Relighting

Chapter 4 established a rendering model that can relight portraits once suitable physics-based inputs are available. This immediately raises the next question in the thesis: can those intermediate representations be generated directly, rather than first recovered from an already rendered image? In other words, instead of treating relighting as a post-processing step applied after image synthesis, can a generative model be redesigned so that controllable material and illumination representations become part of the generation process itself?

This chapter addresses **RQ3** 1.3 in the 2D setting by studying *rendering-aware text-to-image generation*. The central task is no longer to generate only a visually plausible RGB portrait from text, but to generate an editable intermediate description from which the final image can be rendered under controlled illumination. This setting is more demanding than conventional text-to-image synthesis because the model must produce both realistic appearance and a decomposition that remains useful for later manipulation. It also exposes a limitation of standard diffusion pipelines: by predicting final RGB appearance directly, they entangle content, material, and lighting into one output and therefore make relighting difficult after generation.

To address this limitation, this chapter combines latent diffusion with a neural shader in a two-stage pipeline. The generative model is trained to predict G-buffer-style material maps and related intermediate attributes from text, while a separate rendering stage converts those representations into the final image under user-controlled lighting. In this way, the chapter extends the learned rendering framework of Chapter 4 into a generative setting, and serves as the first demonstration in the thesis that generation and rendering can be separated explicitly within a modern diffusion-based image synthesis pipeline.

5.1 Introduction

The field of text-to-image (T2I) synthesis has made remarkable strides, with diffusion-based models now capable of generating images of stunning photorealism and diversity [132, 129]. Despite this progress, a significant frontier remains: fine-grained control. Precisely manipulating attributes such as geometric pose, physics-based illumination, and intrinsic material properties remains an open challenge. This level of granular control is not merely an academic pursuit but a critical necessity for professional applications in visual effects, product design, and gaming, where assets must be editable and integrable. To address this gap, we propose a novel pipeline that uses natural language as its primary input. Our method is built upon the unique combination of a diffusion model and a physics-based neural shader, designed to enable explicit, multi-level control over a scene's G-buffer representation and lighting.

Recent research has pursued this goal of enhanced controllability. For structural and compositional guidance, a significant body of work has emerged. Frameworks like ControlNet [184] and T2I-Adapter [111] introduce trainable modules that condition the generation process on spatial inputs such as Canny edges, depth maps, or human pose skeletons. Other approaches, such as GLIGEN [90], focus on enabling precise object placement through grounded language and bounding box guidance. Concurrently, a separate line of research has focused specifically on illumination. Techniques such as DiffusionLight [173], which estimates environmental lighting by virtually adding a chrome sphere to a scene, and DiLightNet [121] or LightIt [43], which use radiance or shading maps as conditioning, have demonstrated impressive command over the final lighting environment. These approaches represent significant advancements in imposing user intent onto the generative process.

However, these existing methods are constrained by fundamental limitations inherent in the dominant T2I paradigm. Firstly, their underlying frameworks are typically coupled, inextricably entangling content generation with the final image rendering. This fusion of an object's semantics, geometry, and appearance within a single process severely hinders post-synthesis manipulation, such as relighting an object without altering its texture. Secondly, these methods often rely on text to implicitly guide physical attributes, but natural language suffers from a semantic gap; it lacks the precision to quantitatively define lighting intensity, surface roughness, or exact 3D coordinates, leading to results that are plausible but not physically accurate. Lastly, while methods like ControlNet offer structural guidance, the requisite control signals, such as detailed pose skeletons or segmentation maps, are often non-trivial for a user to create or acquire, posing a significant barrier to practical use.

To overcome these challenges, our work introduces a new paradigm that decomposes the image generation process into distinct texturing and rendering stages. We build

our framework upon a pretrained latent diffusion model (Stable Diffusion [130]) and fundamentally re-engineer its synthesis process. The core of our innovation lies in a custom multi-head Variational Autoencoder (VAE). Its decoder, while sharing intermediate layers, features two separate output heads: an RGB head for standard image reconstruction to maintain the pretrained latent distribution, and a novel G-buffer head specifically trained to generate a rich set of physics-based rendering (PBR) material maps, including albedo, roughness, specular, normals, and depth. During inference, a text prompt guides the diffusion model to produce a latent code, which our G-buffer head then decodes into these material maps. These maps are subsequently fed into a physics-based neural deferred renderer, which synthesizes the final image under any user-specified illumination. This explicit separation of material generation from rendering is what enables powerful and granular control.

Our primary contributions are threefold:

- We propose a novel framework called ShadingFusion that decomposes the text-to-image pipeline into two distinct stages of PBR material synthesis and neural rendering, thereby enabling explicit control over physical attributes like material and illumination.

- We present a latent diffusion model featuring a novel multi-head VAE decoder that synthesizes physics-based material maps (G-buffers) from text prompts, shifting the generative target from final pixels to intermediate physical properties.

- We introduce CelebA-PBR-Text, a new large-scale dataset of human faces with paired text descriptions, PBR material maps, and estimated illumination, created specifically to facilitate research on decomposed generative models.

5.2 Related Works

5.2.1 Conditional Diffusion Models

Conditional diffusion models are now central to generative AI. Foundational works like Stable Diffusion [130] and ControlNet [184] first established powerful text-guided synthesis and fine-grained spatial control. Building on this, the field has progressed towards controlling specific physical attributes. One major line of research has achieved sophisticated illumination and relighting by incorporating geometric and reflectance cues into the model [45, 42]. In parallel, geometry-aware approaches have extended diffusion to the 3D domain, enabling the creation of animatable avatars with free-viewpoint control [81, 120]. More recently, the focus has shifted to the direct synthesis of physics-based materials, with techniques emerging that generate PBR texture maps from text prompts [190, 56]. While

these parallel efforts successfully control individual physical attributes, they typically address only one aspect—such as lighting, geometry, or material—at a time. Our work, in contrast, seeks to unify these specialized control axes into a single, cohesive framework.

5.2.2 Neural Deferred Rendering

In computer graphics, deferred shading postpones intensive lighting calculations to a second pass, processing only visible pixels to optimize performance compared to traditional forward shading [23]. This principle has found new life in the era of neural rendering, a field dedicated to synthesizing imagery by learning from data [150]. The integration of deferred shading into neural networks is part of a broader trend toward creating fully differentiable rendering pipelines, which allow optimization directly from image-based losses. This is a key advantage over many classical methods that involve non-differentiable steps like rasterization. The concept of neural deferred rendering was pioneered by Thies et al. [151], who combined a learnable neural texture with a deferred renderer. Worchel et al. [159] later extended this for 3D reconstruction. However, a critical limitation of these prior works is their failure to model the physical principles of light transport; they learn a mapping to an appearance, not to a physically-plausible shading process. Consequently, they cannot generalize to novel illumination conditions. Our work addresses this gap by designing a neural shader that learns a neural approximation of the physics-based rendering equation. By conditioning our shader on key components of this equation—namely incident light vectors, surface material properties (PBR), and the viewing angle—our model can robustly generalize to new lighting, enabling high-fidelity, interactive relighting.

5.2.3 Multi-modality Large language models.

MLLMs have recently demonstrated powerful capabilities in jointly processing and reasoning about visual and textual information, as exemplified by advanced architectures like DeepSeek-VL2 [161] and recent innovations in chain-of-thought reasoning [157]. This has spurred their application in enhancing the controllability of image synthesis, where they primarily act as intelligent interpreters of user intent. For instance, MGIE [30] utilizes an MLLM to transform ambiguous user requests into explicit editing directives, while the Draw-and-Understand framework [97] leverages MLLMs to interpret a combination of text and user-drawn strokes. In contrast to using MLLMs for refining editing instructions, our work employs one to perform a more foundational task: translating free-form language directly into a structured, physics-based scene description. This structured data, containing explicit parameters for lighting and materials, directly conditions our decomposed pipeline and bridges the semantic gap between intuitive language and physics-based synthesis.

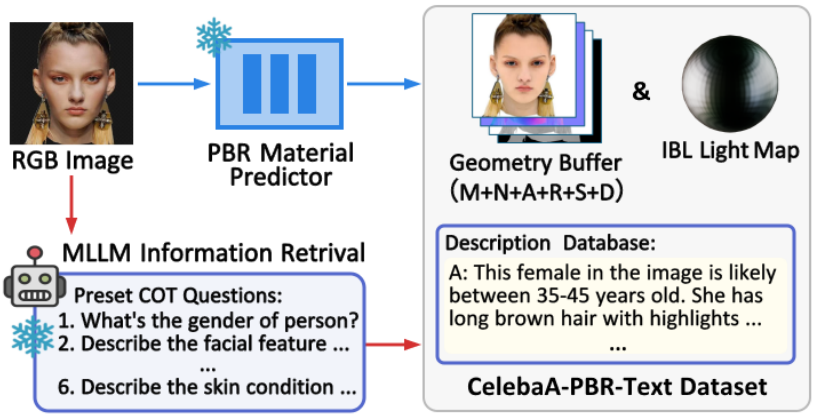

Figure 5.1: Data Collection: PBR Information Acquisition.

5.3 Method

5.3.1 Task Definition

We formulate the task in this chapter as *rendering-aware text-to-image generation*. Let p denote a natural-language prompt, optionally enriched into a structured prompt $\tilde{p}$ by an MLLM. The desired output is not only a final RGB image I_{rgb}, but also a set of intermediate maps

$$M = \{A, R, S, D, N, K\},$$

where A, R, S, D, N, and K denote albedo, roughness, specular reflectance, depth or view coordinates, normals, and foreground mask, respectively. These intermediate maps should be sufficiently consistent and physically meaningful that they can be rendered under a chosen illumination condition to produce the final image.

More formally, the intended mapping can be written as

$$z = F_\theta(\tilde{p}), \qquad \hat{M} = D_\phi(z), \qquad \hat{I}_{rgb} = R_\psi(\hat{M}, L),$$

where F_θ is the text-conditioned latent diffusion model, D_ϕ is the decoder that predicts the intermediate material representation, R_ψ is the neural renderer, and L is the user-specified illumination. The method therefore separates generation into two stages: first predict an editable representation from text, then render that representation into the final RGB appearance. The overall pipeline is shown in Figure 5.2. At a high level, the chapter is organized into three parts: data acquisition for paired text and PBR supervision, a shading-based diffusion model that predicts the intermediate representation, and a physics-based neural deferred renderer for illumination-conditioned image synthesis.

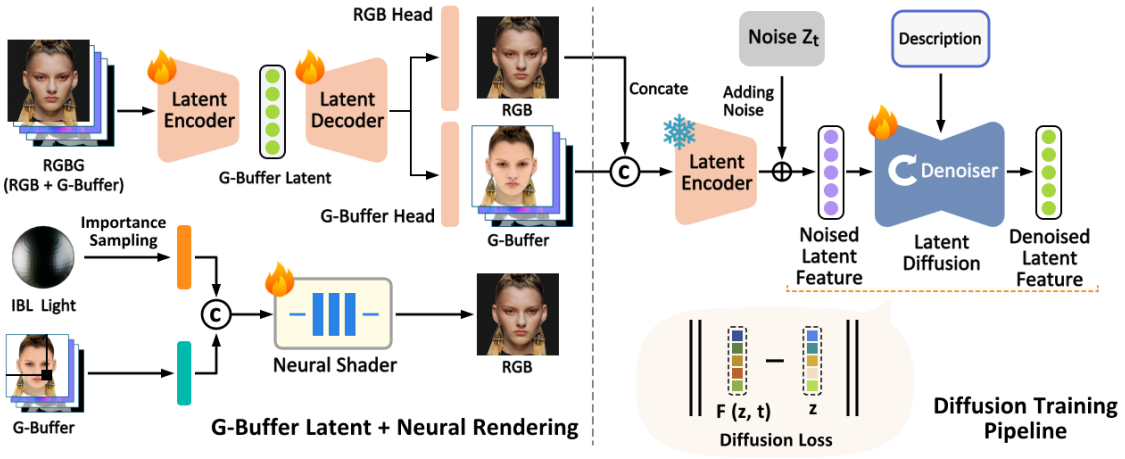

Figure 5.2: The overall pipeline. The diffusion model generates a G-buffer for novel content and uses the neural shader to render the final RGB image.

5.3.2 PBR Information Acquisition

Because collecting PBR materials with precise text descriptions for real-world photos is challenging, there is no public dataset that directly satisfies our requirements. To address this problem, we create our own paired dataset of human faces with text prompts and PBR materials, CelebA-PBR-Text, based on the existing CelebA dataset. Figure 5.1 illustrates the data-acquisition pipeline, which gathers the PBR material textures, scene illumination, and corresponding text descriptions required to train our model.

Specifically, we employ a recent inverse rendering model [77] as the PBR pass to estimate the texture map of physics-based rendering materials and environment illumination of each image in the CelebA dataset. The estimated PBR materials include both geometric information (depth and normals) and textures representing spatially varying material properties (albedo, roughness and specular reflectance). Moreover, we use a multi-modality large language model (MLLM) to extract descriptive information about the human subject in each image. In order to collect the required information, the MLLM performs a Visual Question Answering (VQA) task to answer a fixed set of predetermined questions (Appendix B). We set the output token length to 77 to match the requirement of Stable Diffusion [130], which is used as the backbone in the following process.

5.3.3 Physics-based Latent Diffusion Models

We build our shading-based diffusion model on Stable Diffusion [130] and refer to the resulting system as a physics-based latent diffusion model (PBLDM). The architecture contains three coupled modules: a pretrained VAE encoder $\mathcal{E}$ with a modified decoder $\mathcal{D}_\phi$, a text-conditioned latent U-Net denoiser $\mathcal{F}_\theta$, and a physics-based neural renderer. The overall objective is to keep the latent space compatible with Stable Diffusion while changing

the decoder target from RGB-only reconstruction to joint RGB and G-buffer prediction. We denote the input RGB image as I and the latent representation as Z_0. Training is staged: we first adapt the decoder to reconstruct both RGB and material maps, then fine-tune the denoiser in the same latent space, and finally optimize the renderer using the decoded material outputs.

Architecture Overview. Architecturally, the VAE branch retains the pretrained Stable Diffusion decoder trunk as a shared upsampling backbone and introduces task-specific prediction layers only near the output. This preserves the coarse semantic structure already aligned with the latent distribution while allowing the final prediction stage to specialize for different modalities. The RGB head outputs a standard 3-channel reconstruction used to keep the latent space anchored to the pretrained model. The G-buffer head outputs a 12-channel material stack grouped as albedo, roughness, specular reflectance, depth or view coordinates, normals, and foreground mask. The denoiser remains the standard latent U-Net from Stable Diffusion: it takes a noisy latent, a timestep embedding, and text conditioning, and predicts the noise residual in the same latent space. The renderer is attached after decoding and converts the predicted material maps into the final RGB image under controllable illumination. In other words, the off-the-shelf components are the pretrained VAE encoder, the shared decoder backbone, and the latent diffusion U-Net, while the task-specific additions of this chapter are the G-buffer prediction head, the staged training scheme, the MLLM prompt-enhancement step, and the renderer attachment that converts decoded materials into relightable RGB output.

Multi-head VAE for Facial Materials Decoding. Given an RGB image and its G-buffer I_{RGBG}, our work necessitates the simultaneous reconstruction of the input image and multiple material textures. Specifically, we encode I_{RGBG} with a VAE encoder $\mathcal{E}$ to get the latent feature Z_0:

$$Z_0 = \mathcal{E}(I_{RGBG}) \tag{5.1}$$

We propose a multi-head VAE decoder inspired by previous work [130] to decode the latent representation Z_0, reconstructing the input image $\hat{I}$ and the PBR materials through an RGB head and a G-buffer head, respectively. Within the overall pipeline shown in Figure 5.2, the decoder first passes Z_0 through a shared decoder trunk $\mathcal{D}_{shr}$ and then branches into two modality-specific heads. The shared trunk is responsible for recovering coarse semantic and geometric structure, whereas the output heads specialize in appearance reconstruction and material prediction. This separation reduces cross-task interference while keeping both outputs aligned to the same latent representation. The overall procedure

is defined by Equation 5.2:

$$\begin{aligned} \hat{I} &= \mathcal{D}_{rgb}(\mathcal{D}_{shr}(Z_0)) \\ \hat{A}, \hat{R}, \hat{S}, \hat{D}, \hat{N}, \hat{M} &= \mathcal{D}_{gbuf}(\mathcal{D}_{shr}(Z_0)) \end{aligned} \tag{5.2}$$

where $\mathcal{D}_{shr}$ denotes the shared decoder layers, $\mathcal{D}_{rgb}$ and $\mathcal{D}_{gbuf}$ are the RGB and G-buffer heads, and the two branches preserve the same output resolution while producing different channel groups. Here, $\hat{I} \in [0,1]^3$ is the predicted RGB image; $\hat{D} \in [0,1]^3$, $\hat{N} \in [0,1]^3$, and $\hat{M} \in [0,1]^1$ are the predicted depth or coordinate map, normal map, and mask, respectively; and $\hat{A} \in [0,1]^3$, $\hat{R} \in [0,1]^1$, and $\hat{S} \in [0,1]^1$ are the albedo, roughness, and specular texture maps. We normalize the ranges of $\hat{D}$ and $\hat{N}$ to stabilize training.

The multi-head VAE decoder is trained by minimizing the objective functions (Equation 5.3) via optimizing the neural network parameters ϕ:

$$\underset{\phi}{\operatorname{argmin}}\, \mathcal{L}_{total}(\mathcal{D}_\phi(\mathcal{E}(I_{RGBG})), I_{GT}) \tag{5.3}$$

$\mathcal{E}$ in Equation 5.3 is the pretrained VAE encoder [130], whose parameters remain frozen during training. This prevents drift in the latent distribution and makes the subsequent denoiser fine-tuning more stable. I_{GT} denotes the corresponding supervision collected in the previous phase. The total loss is defined in Equation 5.4; empirically, we set λ to 0.5 for stable convergence.

$$\begin{aligned} \mathcal{L}_{total} = \mathcal{L}_{rgb} + \mathcal{L}_{normal} + \mathcal{L}_{material} \\ + \mathcal{L}_{coords} + \lambda_1 \mathcal{L}_{mask} + \lambda_2 \mathcal{L}_{kl} \end{aligned} \tag{5.4}$$

More specifically, $\mathcal{L}_{total}$ combines modality-specific supervision terms. We use Mean Squared Error (MSE) for RGB reconstruction, view coordinates, and the material channels albedo, roughness, and specular reflectance. For compact notation, we concatenate these material maps along the channel dimension as $\hat{M} = \mathbf{concat}(\hat{A}, \hat{R}, \hat{S})$, as shown in Equation 5.5:

$$\begin{aligned} \mathcal{L}_{rgb} &= \frac{1}{N_b} \sum_{i=1}^{N_b} \|\hat{I} - I\|_2 \\ \mathcal{L}_{coords} &= \frac{1}{N_b} \sum_{i=1}^{N_b} \|\hat{C}_v^i - C_v^i\|_2 \\ \mathcal{L}_{material} &= \frac{1}{N_b} \sum_{i=1}^{N_b} \|\hat{M}_i - M_i\|_2 \end{aligned} \tag{5.5}$$

We use a cosine-similarity loss (Equation 5.6) to supervise surface-normal prediction. For the foreground mask, we use a combined loss consisting of BCE, Dice, and boundary

terms (Equation 5.7).

$$\mathcal{L}_{normal} = 1 - \frac{\sum \hat{N} \cdot N}{\|\hat{N}\|\|N\|} \tag{5.6}$$

$$\begin{aligned} \mathcal{L}_{BCE} &= -\frac{1}{N}\sum_{i=1}^{N}[M_i \log(\hat{M}_i) + (1 - M_i) log(1 - \hat{M}_i)] \\ \mathcal{L}_{DICE} &= 1 - \frac{2\sum M_i \cdot \hat{M}_i}{\sum M_i + \sum \hat{M}_i} \\ \mathcal{L}_{Boundary} &= \int_{\Omega} |\nabla P| dx, \;\; P = Softmax(\hat{M}) \end{aligned} \tag{5.7}$$

The KL loss maps the learned latent distribution to a standard normal distribution so that the latent can be sampled reliably during diffusion training, as defined in Equation 5.8, where $\mathcal{P}$ is the learned distribution.

$$\mathcal{L}_{kl} = \mathrm{KL}(\mathcal{P}||\mathcal{N}) \tag{5.8}$$

After training, our multi-head VAE can encode the PBR materials as well as output its variational latent feature Z_0 for the following diffusion-based generative training.

Facial Material Synthesis. We employ a latent diffusion model (LDM) operating in the latent space of the trained VAE. As shown in Figure 5.2, an input RGB image I is first mapped into the VAE latent space via $Z_0 = \mathcal{E}(I)$.

The forward process of diffusion model is fixed and adds noise to the latent feature iteratively (Equation 5.9):

$$q(Z_t|Z_{t-1}) = \mathcal{N}(Z_t; \sqrt{\alpha_t} Z_{t-1}, (1 - \alpha_t)\mathbf{I}) \tag{5.9}$$

where $t = 1...N$ denotes the denoising step and the variance α_t defines the noise schedule [53], the noisy samples Z_t in any t step are obtained with the standard Gaussian reparameterization (Equation 5.10):

$$Z_t = \sqrt{\bar{\alpha}_t} Z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}) \tag{5.10}$$

We follow the standard noise-prediction training strategy for diffusion models, where a U-Net denoiser $\mathcal{F}_\theta$ predicts the added noise ϵ at each timestep under text conditioning, minimizing the following objective function (Equation 5.11):

$$\underset{\theta}{\operatorname{argmin}}\, \mathcal{L}(\mathcal{F}_\theta(t, x), \epsilon) \tag{5.11}$$

where $\mathcal{L}$ is the Mean Squared Error (MSE) loss, x is the conditioning signal (text prompts

in our case), and ϵ is the noise added to Z_0 at timestep t. Importantly, the denoiser architecture itself is not replaced; instead, it is fine-tuned so that its output latents remain compatible with the modified decoder and therefore decode into both realistic RGB content and physically meaningful G-buffer maps.

During inference, Z_t is sampled from the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The reverse diffusion process then iteratively denoises this latent code, progressively synthesizing novel features that adhere to the target distribution. Finally, a trained decoder translates the synthesized latent representations into physics-based rendering (PBR) materials.

Fine-grained Synthesis Control with MLLM Guidance. To improve text-guided material generation, we design a training-free prompt enhancement procedure that leverages a multi-modal large language model (MLLM). Concretely, we define a facial-description template covering appearance, structure, and illumination cues. The MLLM analyses the user prompt, fills in missing fields, and rewrites underspecified descriptions into a complete prompt that remains within the token budget of the diffusion backbone. This improves the conditioning signal without modifying the diffusion architecture itself.

5.3.4 Physics-based neural deferred rendering

We draw inspiration from the physics-based neural deferred rendering approach in [49] to implement our rendering procedure. The goal is to solve the rendering equation in Equation 5.12 from the synthesized G-buffer textures using a learnable but physically conditioned renderer.

$$L_o(\mathbf{v}) = \int_\Omega F(\mathbf{v}, \mathbf{l}) L_i(\mathbf{l}) \langle \mathbf{n} \cdot \mathbf{l} \rangle d\mathbf{l} \tag{5.12}$$

where $L_o(\mathbf{v})$ is the outbound radiance leaving in direction $\mathbf{v}$; it is the integral of the incident light $L_i(\mathbf{l})$ from every possible direction $\mathbf{l}$ across the hemisphere Ω, centered around the surface normal $\mathbf{n}$. $F(\mathbf{v}, \mathbf{l})$ is the Bidirectional Reflectance Distribution Function (BRDF) describing how the surface reflects light.

The classical renderer in [49] assumes forward-rendering inputs, whereas our G-buffer supervision is obtained from an inverse-rendering pipeline and therefore contains systematic estimation error. A learned renderer is better suited to this setting because it can absorb bias in the estimated materials while still remaining conditioned on physically meaningful variables.

$$\int_\Omega f_\theta(\mathbf{a}, \mathbf{n}, \mathbf{s}, \mathbf{r}, \mathbf{v}, L_i(\mathbf{l}) \langle \mathbf{n} \cdot \mathbf{l} \rangle) d\mathbf{l} \in [0, 1]^3 \tag{5.13}$$

Architecturally, we reuse the two-stage neural shader design introduced in Chapter 4 rather than introducing a separate black-box renderer. After positional encoding of the local material attributes and lighting directions, a diffuse branch predicts a view-independent feature, and a second view-dependent branch combines this feature with the outbound direction to produce the final RGB contribution for each sampled incident ray. In this

way, the renderer keeps the same physically motivated factorization as the Chapter 4 shader while now operating on G-buffers predicted by the generative model. The neural rendering process in Figure 5.2 is defined by Equation 5.13, where f_θ is parameterized by θ, $\mathbf{a}, \mathbf{n}, \mathbf{s}, \mathbf{r}$ are the albedo, normal, specular, and roughness properties of one pixel, $\mathbf{v}$ and $\mathbf{l}$ are the outbound and inbound directions, respectively, and $L_i(\mathbf{l})$ is the incident radiance from direction $\mathbf{l}$. In our implementation, the renderer is trained jointly with the VAE branch, which increases GPU memory usage. To improve efficiency, we replace the uniform sampling strategy in [49] with Monte Carlo importance sampling [153]. The approximation is defined in Equation 5.3.4:

$$L_o(x, \omega_o) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{L_i(\omega_i) f_r(x, \omega_i, \omega_o) |cos\theta_i|}{p(\omega_i)}, \ \omega_i \sim p(\omega)$$

where $L_o(x, \omega_o)$ is the outbound radiance, $L(\omega_i)$ is the incident radiance along ω_i direction, $f_r(x, \omega_i, \omega_o)$ is the bidirectional reflectance function (BRDF), $p(\omega_i)$ is the probability density function (PDF) used by importance sampling.

5.4 Experiments

To evaluate the model, we design three experiments covering generation, content editing, and relighting, each compared against multiple baselines. We first describe the experimental setup, then present qualitative and quantitative comparisons, followed by an ablation study and an application example.

5.4.1 Implementation Details

We use pretrained Stable Diffusion models (SD 1.5, SD XL, SD 3.5, and Flux.1.0-dev) as the backbones in all experiments. The training schedule follows the modular design of the architecture. First, we fine-tune each model's VAE decoder to learn the additional G-buffer branch while keeping the encoder fixed. Second, we train the neural renderer from scratch using the decoded material outputs. Third, we fine-tune the latent U-Net denoiser with the augmented prompts so that sampled latents decode into controllable materials rather than RGB images alone. We use the AdamW optimizer with learning rates of 5e-6 for diffusion-model training and 5e-5 for neural-renderer training. The VAE and neural renderer are trained for 40 epochs, and the U-Net is trained for 20 epochs. The overall training process runs on a single RTX 4090 GPU and converges within roughly 40 hours.

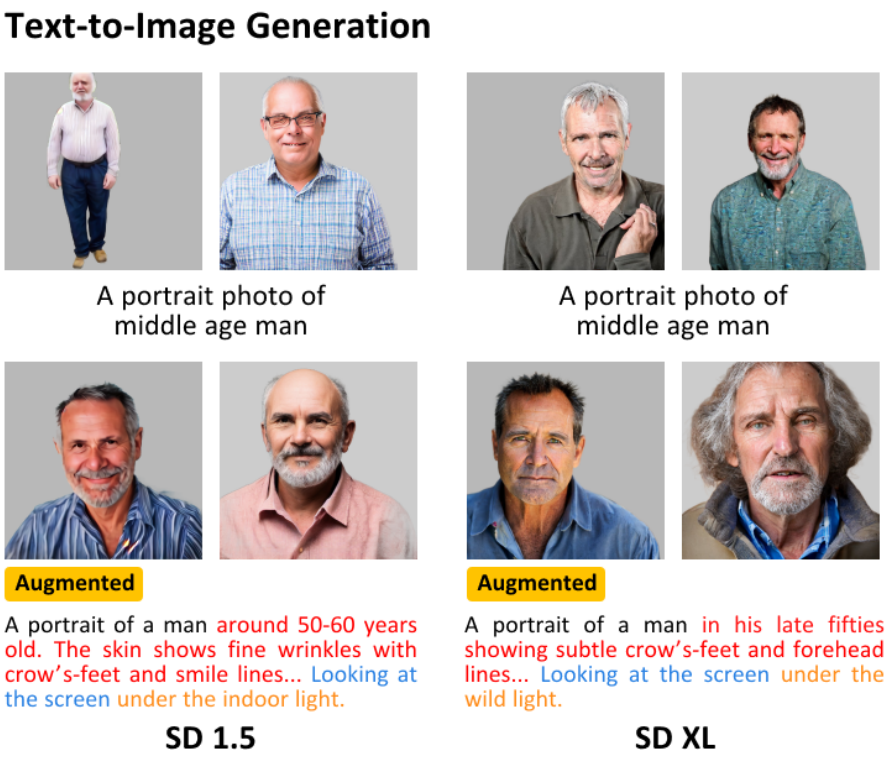

Figure 5.3: Comparison of text-to-image generation results. The red, blue, and orange text highlight the augmented appearance, structure, and illumination descriptions, respectively.

5.4.2 Qualitative Evaluation

The generation experiment evaluates the quality of text-to-image generation for portrait photos. Our work builds upon several pretrained diffusion models. Because SD 3.5 and Flux.1.0-dev introduce large-scale T5 text encoders that support up to 256 tokens, prompt quality is easier to improve through long-form descriptions in those systems. We therefore focus on evaluating the effectiveness of our MLLM enhancement method on SD 1.5 and SD XL, which accept at most 77 input tokens.

Figure 5.3 illustrates the text-to-image generation results of our method and the baseline models (SD 1.5, SD XL). The top row shows generation results from the basic prompt, while the bottom row shows the results with the enhanced prompt. The augmented prompt clearly improves image quality. Our MLLM augmentation pipeline formulates a standard prompt structure that includes appearance, structure, and illumination information. It fills in missing parts of the basic input and adds detail where the original phrasing is vague. For example, a generic middle-aged description in the basic prompt is replaced by a more specific age cue in the augmented prompt.

Next, we compare relighting results against several state-of-the-art baseline models, including SD 3.5 [27], Flux [82], and IC-Light [183], which are widely used for illumination editing. These comparators are appropriate because SD 3.5 and Flux test direct generative

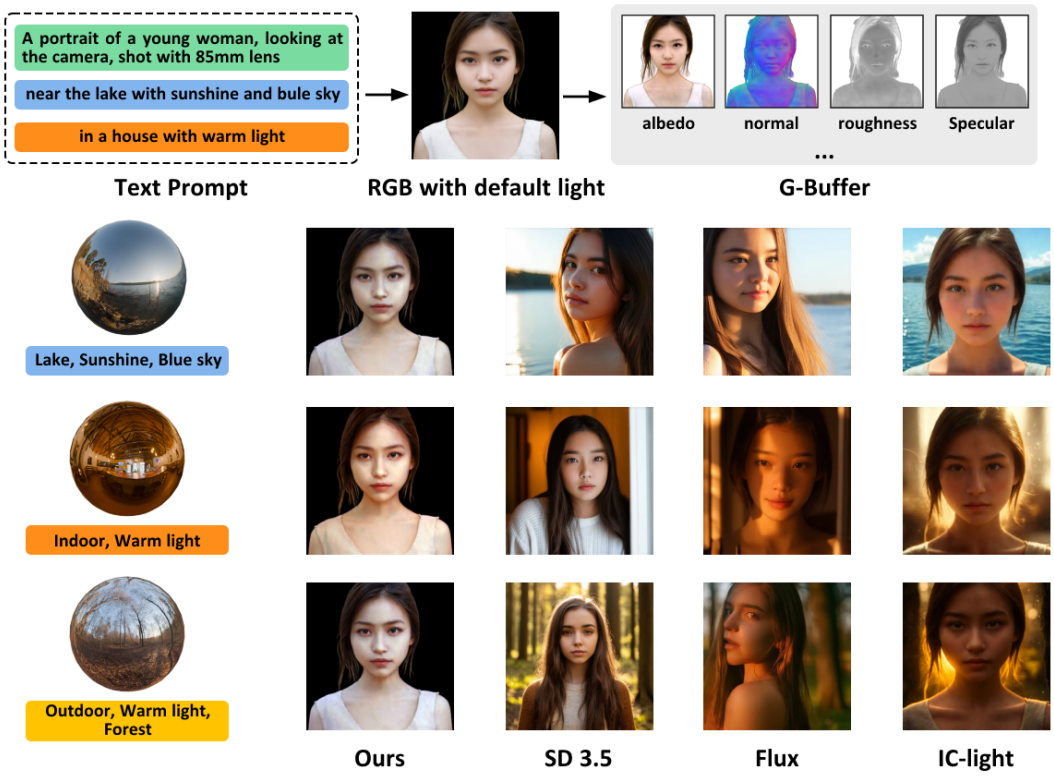

Figure 5.4: Qualitative comparison of the relighting experiment. Our method preserves content identity more consistently than other text-to-image generation models and demonstrates stronger photorealism than the other generative baselines.

relighting, whereas IC-Light tests identity-preserving image-to-image lighting control. Figure 5.4 shows the relighting result of each approach. Compared with other text-to-image methods, our approach first generates the G-buffer from text input and then performs relighting directly on the generated G-buffer, thereby preserving object identity under new lighting conditions. In contrast, SD 3.5 and Flux apply relighting through a generative process, which causes variations in both content and pose across samples. IC-Light maintains content identity more effectively, but its illumination changes remain less consistent than those of our method.

5.4.3 Quantitative Evaluation

We also compare generation and relighting performance quantitatively. Because there is no pixel-wise ground truth for open-ended generation, we evaluate each approach using Inception Score (IS) [133], Fréchet Inception Distance (FID) [50], and Kernel Inception Distance (KID) [8]. Higher IS and lower FID or KID indicate better agreement between the generated distribution and the target dataset distribution.

We first evaluate text-to-image generation against two baseline models, SD 1.5 and SD XL, to test the effectiveness of the MLLM augmentation pipeline. Table 5.1 reports

the quantitative comparison between our text-to-image pipeline built on top of SD 1.5 and SD XL and the two vanilla Stable Diffusion backbones on both the CelebA-PBR-Text (CP) and FFHQ (FP) benchmarks. Because CelebA and CelebA-PBR-Text share image content, we use the RGB images from the evaluation split of CelebA-PBR-Text throughout the following experiments. We generate 2,000 samples for each method, using the basic prompts for the vanilla models and the augmented prompts for our models, and compute the similarity metrics against the same number of dataset samples.

As shown in Table 5.1, both of our models substantially outperform their respective base models. Ours(SD 1.5) achieves an IS of 2.910/2.909 on CP/FP, compared with 2.013/1.901 for SD 1.5. Similarly, Ours(SD XL) yields 2.935/2.926 versus 2.342/2.316 for SD XL. This indicates that our modifications improve the diversity and coverage of the generated samples. Our method also delivers much lower FID scores, confirming higher perceptual quality. On CP, Ours(SD 1.5) reduces FID from 2.103 to 0.879, and Ours(SD XL) from 1.527 to 0.893. A similar trend appears on FP, where FID improves from 1.945 to 1.294 for SD 1.5 and from 1.526 to 1.132 for SD XL. We also obtain the lowest KID values overall. In particular, Ours(SD 1.5) reaches only 0.036/0.101 on CP/FP, which is a substantial reduction relative to the baselines. These results show that the pipeline improves sample fidelity and consistency regardless of whether SD 1.5 or SD XL is used as the backbone.

	IS ↑		FID ↓		KID ↓	
	CP	FP	CP	FP	CP	FP
SD 1.5	2.013	1.901	2.103	1.945	0.324	0.433
SD XL	2.342	2.316	1.527	1.526	0.165	0.147
Ours (SD 1.5)	2.910	2.909	0.879	**1.294**	**0.036**	0.101
Ours (SD XL)	**2.935**	**2.926**	**0.893**	1.132	0.051	**0.084**

Abbreviations. CP: CelebA-PBR-Text; FP: FFHQ.

Table 5.1: Quantitative evaluation of text-to-image generation on the CelebA-PBR-Text and FFHQ benchmarks, comparing the proposed decomposed pipeline against vanilla Stable Diffusion backbones.

We also evaluate the relighting experiment quantitatively in Table 5.2 against three state-of-the-art baselines, SD 3.5, Flux 1.0, and IC-Light, on the CelebA-PBR-Text (CP) and FFHQ (FP) datasets. SD 3.5 and Flux achieve strong IS scores (around 2.7) but show weaker FID and KID values, indicating that direct generative relighting does not preserve image statistics as reliably as our decomposed pipeline. IC-Light achieves the highest IS (2.812/2.983) but still produces moderate FID (1.032/1.107) and KID (0.105/0.155). Our ShadingFusion method achieves the lowest KID and near-best FID among the compared methods. These results support two claims: first, direct relighting in G-buffer space preserves identity more consistently than purely generative relighting; second, combining the neural renderer with importance sampling yields a strong balance between sampling

efficiency and perceptual fidelity. The role of each component is further analysed in the following ablation study.

	IS ↑		FID ↓		KID ↓	
	CP	FP	CP	FP	CP	FP
SD 3.5	2.748	2.734	0.991	0.986	0.111	0.189
Flux 1.0	2.671	2.785	**0.982**	0.991	0.113	0.177
IC-Light	2.812	**2.983**	1.032	1.107	0.105	0.155
Ours (GGX)	1.225	1.203	2.506	2.630	0.559	0.661
Ours (BP)	1.130	1.201	2.412	2.531	0.638	0.576
Ours (US)	2.811	2.621	0.936	0.972	0.105	0.158
Ours (IS)	**2.820**	2.633	0.994	**0.971**	**0.103**	**0.153**

Abbreviations. US: uniform sampling; IS: importance sampling [153]; GGX: Trowbridge-Reitz GGX model [154]; BP: Blinn-Phong shading model [9]; CP: CelebA-PBR-Text; FP: FFHQ.

Table 5.2: Quantitative evaluation of the relighting experiment on the CelebA-PBR-Text and FFHQ benchmarks, comparing direct generative relighting baselines with classical-shading and neural-rendering variants of our pipeline.

5.4.4 Ablation Study

We conducted an ablation study to evaluate the importance of the different components of the ShadingFusion pipeline following the same protocol as in Section 5.4.3. Ours (GGX) and Ours (BP) in Table 5.2 replace the neural renderer with classical GGX [154] and Blinn-Phong [9] shading, respectively. When fixed GGX or Blinn-Phong shading is used without importance sampling, IS drops dramatically, reflecting inefficient exploration of the lighting space, while FID and KID degrade strongly. Ours (US) replaces importance sampling with Monte Carlo uniform sampling. This recovers competitive FID and KID, but IS remains below the best-performing models. Ours (IS) uses Monte Carlo importance sampling [153] in the neural renderer, restoring IS to the level of the strongest baselines while achieving the lowest KID and near-best FID. We therefore select the neural renderer with importance sampling as the final rendering design.

5.5 Conclusion

In this chapter, we introduced ShadingFusion, a decomposition-based text-to-image pipeline that shifts the generative target from final RGB pixels to editable physics-based scene attributes. The experiments indicate that this formulation preserves image quality while enabling a capability that standard text-to-image models do not natively provide: relighting after generation with stronger identity consistency across lighting changes. The chapter also

introduced CelebA-PBR-Text as a paired dataset of portraits, estimated PBR attributes, and text descriptions for studying this problem.

The method still has important limitations. The generated representation is a 2D image-plane G-buffer rather than a full 3D scene description, so viewpoint control remains limited compared with explicit 3D methods. The decomposition quality is also bounded by the inverse-rendering estimates used for supervision and by the degree to which the neural shader respects physical constraints such as energy conservation. These limitations motivate the move to 3D Gaussian scene generation in Chapter 6.

Chapter 6

DiffGSPBR: Physics-Based Rendering of Generative Gaussian Splats for Decomposed 3D Synthesis

The G-buffer representation in Chapter 5 enabled relighting but still tied the generated content to a single image plane. Although that decomposition made illumination editable after generation, it did not provide a true scene representation: geometry remained implicit in image space, viewpoint changes were limited, and the generated result could not be manipulated as a consistent 3D object. The next step in the thesis is therefore to ask whether the same rendering-aware principle can be extended from relightable 2D images to relightable 3D scenes.

In the 3D setting, the model must not only synthesize visually plausible content, but also maintain consistency across views while separating geometry, material, and illumination into components that remain useful for later editing. A successful method must therefore achieve several goals simultaneously: generate coherent 3D structure, preserve photorealistic appearance, support relighting under new environments, and allow viewpoint changes without collapsing the scene representation. These requirements go beyond standard text-to-3D or image-to-3D generation, where appearance is often learned as a fused output rather than as an editable decomposition.

This chapter addresses **RQ3** 1.3 in the 3D setting by integrating generative Gaussian splatting with physics-based deferred rendering. The resulting framework, *DiffGSPBR*, combines a generative 3D Gaussian backbone with decomposition modules for material and illumination, together with a rendering stage that reconstructs the final RGB appearance from those predicted scene attributes. In this way, the chapter extends the thesis argument from controllable 2D image synthesis to editable 3D scene generation, and studies whether rendering-aware decomposition can be introduced into a generative 3D pipeline without sacrificing generation quality.

6.1 Introduction

Chapter 5 demonstrated that intermediate physics-based representations can decouple generation from rendering, but the resulting G-buffer remains a 2D representation and therefore cannot provide true viewpoint control under standard model-view-projection transformations. In this chapter, we move to a 3D generation pipeline that synthesizes a scene with decomposed geometry, materials, and illumination, enabling fine-grained control over both camera pose and relighting. Specifically, the complete representation of 3D scene can be divided into three parts, geometry, material, and illumination. Thus, synthesizing a novel 3D scene requires jointly generating these three components. Our method demonstrates this capability in Figure 6.1, which shows geometry synthesis, material decomposition, and relighting from a single image or prompt.

Diffusion based generative models [53, 129] have made great advances in generating 3D geometry with pre-computed color. Some works train conditional 3D generative models directly on datasets of various 3D representations [114, 63, 11, 46, 180, 96] or performing the rendering supervision by using differentiable rendering techniques [2, 68, 145]; other works reconstruct the implicit 3D representation by utilizing pretrained multi-view diffusion models [86]. Although these works can conditionally generate 3D contents with high quality and view consistency, none of them can decompose the generated 3D scene to enable further editing function for the material and illumination. Other material generative models [189, 168] focus on the generation of physics-based materials conditioned by the mesh or textual input, rather than generating the decomposed representation of a 3D scene.

Inverse rendering techniques recover the physical parameters of a given 3D scene including the geometry, material and illumination from captured images [5]. A common method is to initialize the unknown attributes of a 3D scene as the trainable parameters, and then are optimized by the reconstruction loss between the posed rendering image and ground truths. Recent research of inverse rendering has adopted the Neural Radiance Fields (NeRF) [110] to reconstruct the geometry of the 3D scene, but face challenges in modeling the physical attributes of scenes as its implicit representation. More recently, 3D Gaussian Splatting (3DGS) [74] has emerged as a popular and efficient 3D representation. It offers a lightweight parameterization that explicitly encodes both geometric structure and appearance through a set of spatially distributed Gaussians. Many studies [62, 138, 95, 33, 160, 15] attempt to use 3DGS in inverse rendering framework, successfully decoupling color from the interaction between lighting and surface materials by incorporating physics-based rendering (PBR).

Drawing inspiration from generative Gaussian splatting models [96] and deferred 3D Gaussian rendering [160, 15], we propose DiffGSPBR, a novel generative gaussian splatting model that integrates physics-based rendering (PBR) in a deferred 3D Gaussian rendering pipeline to produce 3D scenes with material and lighting decomposed. In summary, our

Figure 6.1: Our method enables end-to-end synthesis, decomposition, and physics-based rendering of 3D Gaussian splats (3DGS). Given a single image or text prompt, we generate 3DGS geometry and predict surface attributes like normal, albedo, and roughness. A Gaussian deferred renderer produces high-quality novel view synthesis and the deferred shading under the environment lighting. Central to our framework is a self-supervised decomposition pipeline built on a pretrained generative prior, combined with a deferred rendering pipeline that disentangles appearance and illumination.

contributions are:

- We present DiffGSPBR, a self-supervised 3D synthesis decomposition framework which generates 3D Gaussian splats with decoupled physics-based materials and illumination.
- We propose the Gaussian material score distillation by using a pretrained material diffusion model, mitigating the metal degradation during the PBR material estimation.
- Extensive experiments demonstrate the superior performance of the proposed method, and ablation studies are conducted to analyze the effectiveness of each design choice.

6.2 Related Works

6.2.1 3D Generative Models

Recent advances in diffusion-based generative models [53, 142] have sparked growing interest in 3D content generation. Existing methods can be broadly categorized into three types: Native 3D, rendering-based, and reconstruction-based. Native 3D models directly train diffusion networks on explicit or implicit 3D representations, such as voxels [58, 112, 149], point clouds [103, 114], implicit fields [100, 182, 89], triplanes [14, 141, 156], and 3D Gaussians [46, 180]. While offering 3D consistency, these methods rely on large 3D datasets

and cannot benefit from pretrained 2D models. Rendering-based methods optimize 3D content under 2D supervision using differentiable rendering [110], as seen in works like HoloDiffusion [68, 67], GIBR [1], and DMV3D [167]. Though data-efficient, they often entangle geometry and appearance, limiting interpretability. Reconstruction-based methods synthesize multi-view images via 2D diffusion, then reconstruct 3D representations [155, 137, 139, 41]. Studies including Instant3D [86] and LRM-based pipelines [55, 152], as well as VolSDF [170] and FlexiCubes [135, 136] suffer from reliance on view consistency and fragile two-stage designs. Despite their progress, existing paradigms often suffer from slow optimization, high data requirements, and limited rendering efficiency. These limitations have motivated recent interest in 3D 3DGS [74], a lightweight and explicit 3D representation that supports real-time differentiable rendering and compact geometry modeling.

6.2.2 Generative 3D Gaussian Splatting Models

Generative models based on 3DGS [74] offer a promising direction for efficient 3D content synthesis. Diffusion-based approaches such as GSGEN [18] and DiffSplat [96] leverage pretrained 2D diffusion models to directly generate 3D Gaussian primitives with high visual fidelity. Other pipelines integrate Score Distillation Sampling (SDS) [107] with 3DGS initialization to accelerate training and improve generalization. GaussianDreamer [172] and AGG [163] use 3D diffusion or amortized networks for fast generation, while DreamGaussian [148] and LucidDreamer [94] constructs full 3D scenes by progressively densifying lifted point clouds, and improves pseudo-ground truth consistency via interval score matching and DDIM inversion. Text2Immersion [117] refines 3DGS in two stages with additional views, and despite these advances, most existing methods still entangle geometry, appearance, and lighting, limiting control and physical interpretability. In contrast, this chapter introduces a physics-based generative framework that explicitly decomposes material and illumination during the 3DGS synthesis, enabling editable and semantically consistent 3D scene generation.

6.2.3 Physics-Based Scene Decomposition and Inverse Rendering

Physics-based scene decomposition aims to separate intrinsic scene properties such as geometry, materials, and illumination from observed appearance. Classical intrinsic image decomposition methods [92, 106] estimate reflectance and shading from a single image, but lack 3D consistency and cannot generalize to view-dependent effects. Inverse rendering techniques, on the other hand, recover scene-level physical attributes by optimizing the reconstruction loss between rendered and captured images [5]. Typically, these methods initialize geometry, materials, and lighting as trainable parameters, and jointly refine them to match posed ground truths. Recent work has incorporated NeRF [110] into inverse

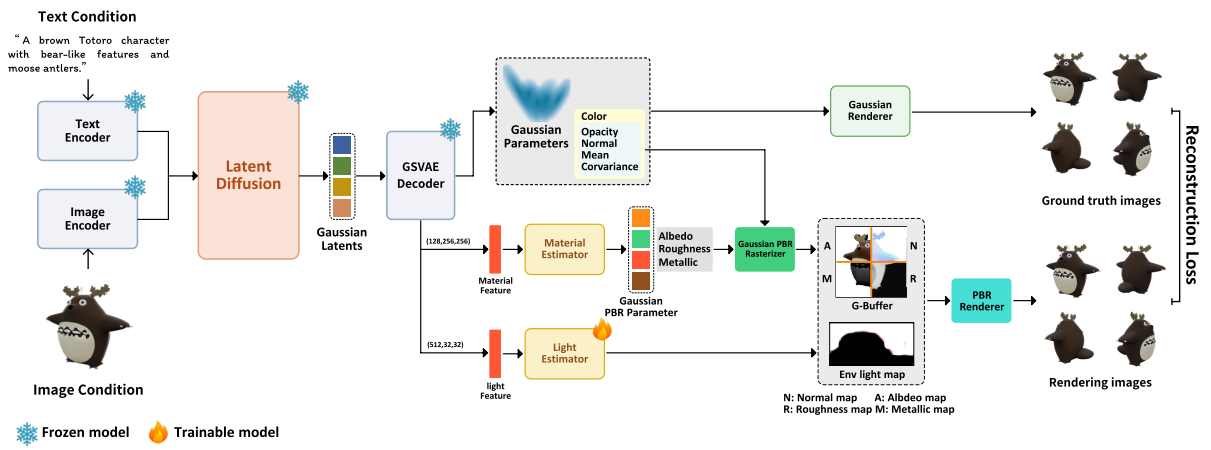

Figure 6.2: Our DiffGSPBR Framework: Generating, Decomposing, and Rendering 3D Gaussian Splatting in Three Stages.

rendering, achieving compelling geometry recovery, but struggles to explicitly disentangle materials and lighting due to its implicit representation. Building on 3DGS representations, a growing number of scene-specific inverse rendering methods [62, 138, 95, 33, 160, 15] combine it with PBR to separately model lighting and surface properties. However, these pipelines rely on captured multi-view images or dense video sequences as input, and lack generalization beyond specific scenes. In contrast, we propose a unified generative framework that incorporates material and illumination decomposition during synthesis, enabling editable and relightable 3D generation.

6.3 Preliminaries

6.3.1 Structured 3D Gaussian Splatting

The standard 3DGS dynamically optimizes a set of Gaussian splats to efficiently represent the geometry of a single scene; however, it is scene specific. Structured 3D Gaussian Splatting [74] assigns Gaussian splats to the 2D image grid to bind each splat to a fixed image pixel. This structured formulation enables learning priors or estimating parameters directly from a neural model. Specifically, each 3DGS primitive is explicitly parametrized by a multivariate Gaussian distribution:

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}, \tag{6.1}$$

where $\mu \in \mathbb{R}^3$ is the mean, representing the position of the Gaussian in 3D space, and $\Sigma \in \mathbb{R}^{3\times3}$ is the covariance matrix, encoding its spatial extent, scale, and orientation. Each Gaussian is further assigned a color $\mathbf{c} \in \mathbb{R}^3$ and an opacity value $\alpha \in \mathbb{R}$, jointly defining its visual appearance. In the rendering process, the structured 3DGS is projected onto the 2D

image plane using standard EWA splatting [193]. Each screen pixel color is obtained by α-blending the projected Gaussian splats sorted by depth, as follows:

$$C = \sum_{i \in \mathcal{N}} T_i c_i \alpha_i, \quad \text{with } T_i = \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{6.2}$$

Here, $\mathcal{N}$ denotes the set of Gaussian splats sorted by depth, $\alpha_i \in [0, 1]$ is the opacity of the i-th splat, and T_i represents the accumulated transmittance, computed as the product of $(1 - \alpha)$ over all preceding splats.

6.3.2 Physics-based Deferred Rendering

PBR is a rendering approach that models the light-surface interaction of an object. PBR frameworks generally adopt microfacet-based reflection models that enforce energy conservation and view-independent material behavior. Practical systems, such as the metallic-roughness workflow, often parameterize materials using albedo, roughness, and metallicity. The deferred rendering is to delay the reflectance calculation until after the rasterization process, only calculate the shading color for the visible points of the screen. Given the position x and normal vector n of a surface point, the outbound light L_o from view direction ω_o can be calculated by the standard rendering function [65]:

$$L_o(\omega_o, x) = \int_\Omega f_r(\omega_o, \omega_i, x) L_i(\omega_i, x)(\omega_i \cdot n) d\omega_i, \tag{6.3}$$

where Ω is the upper hemisphere centered at x and $f_r(\omega_o, \omega_i, x)$ is the bidirectional reflectance distribution function (BRDF), parameterizing the ratio of reflectance between the outbound light L_o on the direction ω_o and inbound light L_i on the direction ω_i based on the material of the position x. We use the widely adopted Cook-Torrance BRDF model [20] in this project. It decomposes reflectance into a diffuse component f_d and a specular component f_s, as shown in Equation 6.4:

$$f_r(\omega_o, \omega_i, x) = f_d(x) + f_s(\omega_o, \omega_i, x) \tag{6.4}$$

The functions f_d and f_s govern, respectively, the diffuse and specular components of the surface reflectance. The diffuse term is given by:

$$f_d(x) = (1 - \mathbf{m}) \frac{\mathbf{a}}{\pi} \tag{6.5}$$

where $\mathbf{a} \in [0, 1]^3$ is the spatially varying albedo and $\mathbf{r} \in [0, 1]$ is the metallic parameter.

The specular term is expressed as

$$f_s(\omega_o, \omega_i) = \frac{D(\mathbf{h};\mathbf{r})F(\omega_o, \mathbf{h};\mathbf{a},\mathbf{m})G(\omega_i, \omega_o, \mathbf{h};\mathbf{r})}{4(n \cdot \omega_i)(n \cdot \omega_o)}, \tag{6.6}$$

where $\mathbf{r} \in [0, 1]$ is the surface roughness. $\mathbf{h} = \frac{\omega_o+\omega_i}{||\omega_o+\omega_i||}$ is the half vector. $D(\mathbf{h};\mathbf{r})$ is the normal distribution function (NDF), $F(\omega_o, \mathbf{h};\mathbf{a},\mathbf{m})$ is the Fresnel term, and $G(\omega_i, \omega_o, \mathbf{h};\mathbf{r})$ is the geometric attenuation.

Combining these, the outgoing radiance L_o at position x for view direction ω_o decomposes into diffuse and specular contributions:

$$L_o(\omega_o, x) = L_d(x) + L_s(\omega_o, x) \tag{6.7}$$

with

$$L_d(x) = \int_\omega f_d L_i(\omega_i, x)(\omega_i \cdot n)d\omega_i \tag{6.8}$$

$$L_s(x) = \int_\omega f_s L_i(\omega_i, x)(\omega_i \cdot n)d\omega_i. \tag{6.9}$$

Here, $L_i(\omega_i, x)$ denotes the incident radiance from direction ω_i, $\mathbf{n}$ is the surface normal, and the integrals are taken over the hemisphere Ω. This formulation cleanly separates energy transport into a Lambertian base modified by metallicity (f_d) and a microfacet-based specular reflection (f_s).

6.4 Method

6.4.1 Task Definition

We formulate the task in this chapter as *editable 3D generation with decomposed scene attributes*. The input can be either a text prompt or a single image. The output is a structured 3D Gaussian scene together with estimated material attributes and environmental illumination, from which RGB images can be rendered under arbitrary viewpoints and lighting. Compared with Chapter 5, the objective is no longer to generate a single relightable image but to generate a relightable and view-consistent 3D scene representation.

Motivated by recent advances in web-scale 3D content generation from diffusion-based generative Gaussian splatting [148, 172, 163, 94], and novel Gaussian-splatting-based inverse-rendering approaches, our target is to decompose the synthesized 3D scene to enable material and illumination editing on generated Gaussian primitives. We present *DiffGSPBR*, a self-supervised framework for generating 3D scenes with decomposed physics-based materials and illumination. As illustrated in Figure 6.2, the pipeline consists of three stages: in the first generation stage, structured 3D Gaussian splats are synthesized

from image or text input and rendered into posed views for self-supervision (Section 6.4.2); in the second decomposition stage, the synthesized scene is disentangled into "pseudo" physics-based material properties and environmental illumination (Section 6.4.3); and in the final rendering stage, a physics-based renderer with deferred shading and differentiable path tracing is used to re-render the scene, supervised by reconstruction loss against the rendered ground truth (Section 6.4.4).

Architecture Overview. The full architecture combines one off-the-shelf generative backbone with two decomposition-specific prediction heads and one rendering module. The generative backbone is a diffusion U-Net plus GS-VAE decoder that synthesizes the Gaussian scene itself. On top of this backbone, we add a local material-estimation head F_{mat} and a global illumination-estimation head F_{light}. This split is deliberate: material should stay spatially aligned with individual Gaussian features, whereas illumination should be shared across all rendered views. The final renderer consumes the predicted material maps, cubemap lighting, depth, and normal buffers to produce the RGB supervision signal used for self-supervised optimization. Thus, the off-the-shelf part of the architecture is the pretrained DiffSplat-style Gaussian generator, while the task-specific additions introduced in this chapter are the material head, the lighting head, and the deferred rendering loop that enforces a physically meaningful decomposition after generation.

6.4.2 Structured 3D Gaussian Parameter Generation

Inspired by recent generative Gaussian splatting methods such as DiffSplat [96], we represent 3D objects using multi-view Gaussian splat grids. Given an image or text prompt, an off-the-shelf diffusion U-Net denoiser generates V_{in} posed latent features in the splat latent space [96]. These latent features are decoded by a GS-VAE into a set of Gaussian primitives $G = \{\mathtt{g}_i\}_{i=1}^{N}$, each aligned to a fixed-resolution 2D grid. In our implementation, this backbone is responsible only for geometry and appearance synthesis; the decomposition modules described next are intentionally attached as lightweight heads rather than replacing the generator itself.

To provide supervision for the subsequent material and illumination estimation, we render V_{out} posed images $\mathbf{I}_{\text{gt}}$ from the structured Gaussian as ground-truth views where V_{out} is the number of output views, $\mathbf{I}_{\text{gt}}$ is the rendered ground truth image. In addition, we extract geometry features including depth ($\mathbf{d}$) and surface normals ($\mathbf{n}$) in this stage. For depth rendering, we follow the approach proposed in GS-IR [95]: the per-pixel depth is computed as the expected depth along a camera ray:

$$d = \sum_{i=1}^{N} w_i d_i, \quad \text{where } w_i = \frac{T_i \alpha_i}{\sum_{j=1}^{N} T_j \alpha_j}, \tag{6.10}$$

where N is the number of sampled points along the camera ray, α_i denotes the opacity of the i-th splat, and T_i is the accumulated transmittance. For normal rendering, we treat normals ($\mathbf{n}$) as attributes associated with each Gaussian and apply α-blending to obtain the final normal map.

6.4.3 Physics-Based Scene Decomposition

In the second stage, we decompose the synthesized 3D scene by estimating PBR materials and illumination.

Material Decomposition The standard 3DGS makes it difficult to associate material attributes, as it is not compatible with the standard microfacet-based BRDF model [4]. To enable PBR within the 3DGS pipeline, we incorporate a deferred rendering technique [160], which performs shading on projected 2D texture maps and bypasses the need for explicit geometry computations.

Specifically, we design a material estimator F_{mat}, a lightweight local prediction head that estimates per-pixel pseudo-material parameters for the Gaussian splats. To capture object-aware information, we take the late-stage feature map X_{late} from the final convolutional layer of the GS-VAE decoder as input. The estimator preserves the spatial layout of this feature map and performs only local feature refinement before the final channel projection, because material parameters should remain aligned with the generated Gaussian geometry rather than be mixed globally across views.

$$g_i = F_{\text{mat}}(x_{\text{late}}) \tag{6.11}$$

As shown in Equation 6.11, F_{mat} predicts material parameters $\mathbf{g}_i \in [0,1]^5$, including albedo $\mathbf{a} \in [0,1]^3$, roughness $r \in [0,1]$, and metallicity $m \in [0,1]$. Operationally, the head acts as a dense per-location regressor: each spatial location in X_{late} is mapped to one material vector, and neighbouring locations interact only through the local receptive field of the head. These per-splat material parameters are then rasterized using standard α-blending to produce pseudo-material texture maps for the subsequent rendering stage.

Illumination Decomposition Unlike material attributes, illumination is a global factor that influences all rendered views. We therefore design a light estimator F_{light} that performs the opposite aggregation pattern to F_{mat}: it discards fine spatial detail and predicts one shared scene-level lighting representation. The estimator takes an early feature map X_{early} from the GS-VAE decoder as input (Equation 6.12). Specifically, given the multi-view latent features V_{in} from the diffusion denoiser, we extract x_{early}, the output of the first convolutional layer, with shape $V_{\text{in}} \times 512 \times 32 \times 32$. We then apply average pooling to suppress high-frequency noise and aggregate the features into a compact global descriptor, from which F_{light} regresses a cubemap light L_{cube} represented by six directional light maps,

as shown in Figure 6.3.

$$L_{cube} = F_{\text{light}}(X_{early}) \tag{6.12}$$

This division of labour between F_{mat} and F_{light} is important for stability. F_{mat} remains attached to late, high-resolution features because those features retain local object structure, while F_{light} uses early, lower-frequency features because illumination should be shared across the full object and across viewpoints.

6.4.4 Physics-Based Gaussian Deferred Rendering

We reconstruct the decomposed scene at this stage to optimize the material and illumination estimators. The renderer is not a new scene generator; instead, it is the supervision module that closes the loop between predicted materials, predicted lighting, and RGB reconstruction. To capture view-dependent illumination accurately, we divide global illumination into direct light L^{dir} and indirect light L^{ind}.

Direct Lighting Modeling We employ image-based lighting (IBL) for the direct-light component. As shown in Equation 6.5, the outgoing radiance can be decomposed into a diffuse component L_d and a specular component L_s. For the diffuse term, we further separate direct and indirect incident light, as shown in Equation 6.13:

$$L_d(x) = (1-m)\frac{a}{\pi}\int_{\Omega}(L_i^{dir}(\omega_i, x) + L_i^{ind}(\omega_i, x))(\omega_i \cdot n)d\omega_i \tag{6.13}$$

Equations 6.14 and 6.15 calculate the irradiance of direct lighting L_i^{dir} in the visible region Ω_{vis} and indirect lighting L_i^{ind} in the occluded region Ω_{occ} of the upper hemisphere ($\Omega_{vis} \cup \Omega_{occ} = \Omega$, $\Omega_{vis} \cap \Omega_{occ} = \varnothing$).

$$L_d^{dir} = (1-m)\frac{a}{\pi}\int_{\omega_{vis}} L_i^{dir}(\omega_i, x)(\omega_i \cdot n)d\omega_i \tag{6.14}$$

$$L_d^{ind} = (1-m)\frac{a}{\pi}\int_{\omega_{occ}} L_i^{ind}(\omega_i, x)(\omega_i \cdot n)d\omega_i \tag{6.15}$$

We then approximate these two integrals using the direct irradiance I_{dir} and indirect irradiance I_{ind}. Here, m and a are the metallicity and albedo introduced in Section 6.3.2, and an occlusion term $O(x)$ approximates visibility, as shown in Equation 6.16.

$$L_d(x) \approx (1-m)\frac{a}{\pi}O(x)I_{dir}(x) + (1-m)\frac{a}{\pi}I_{ind}(x) \tag{6.16}$$

For the specular component L_s, we use the widely adopted split-sum approximation [66]

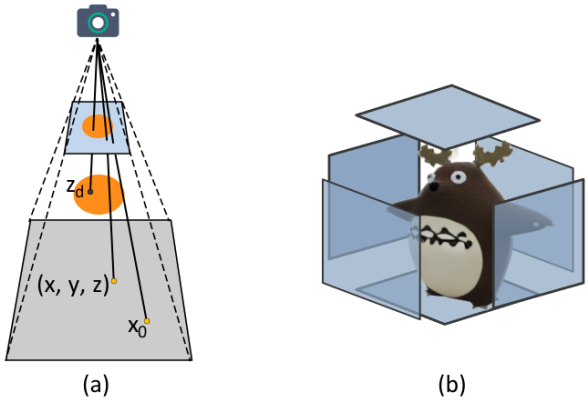

Figure 6.3: Illustration of our cubemap-based indirect lighting estimation. (a) Path tracing is performed from each surface point to evaluate ambient occlusion and visibility. (b) A cubemap is rendered from the surface point by capturing the scene from six orthogonal directions. This cubemap encodes directional occlusion and shading information, enabling efficient visibility-aware integration of indirect lighting.

to handle the intractable integral, as shown in Equation 6.17:

$$\begin{aligned} L_s &= \int_\Omega \frac{DFG}{4(n\cdot\omega_i)(n\cdot\omega_o)} L_i(\omega_i, x)(\omega_i\cdot n)d\omega_i \\ &\approx \underbrace{\int_\Omega \frac{DFG}{4(n\cdot\omega_o)} d\omega_i}_{BRDF\ integral\ R} \underbrace{\int_\Omega L_i(\omega_i) D(\omega_i,\omega_o)(\omega_i\cdot n)d\omega_i}_{Pre-filtered\ envmap\ I_s} \end{aligned} \tag{6.17}$$

where R represents the integral of BRDF under a constant environment light map, which can be precomputed and stored in a 2D look-up table. The second I_s represents the incident irradiance which can also be stored in a pre-filtered environment map with different mip-levels for different roughness values. Hence the rendering result under direct lighting can be written as:

$$L_{dir} = (1-m)\frac{a}{\pi} O(x) I_{dir} + R I_s, \tag{6.18}$$

Indirect Lighting Modeling Indirect lighting, resulting from light bouncing off surfaces, is essential for realistic rendering—especially in occluded or concave regions (Figure 6.4(a)). We estimate this effect by casting rays across the hemisphere aligned with each surface normal to evaluate both occlusion and secondary radiance.

Since 3DGS represents scenes as sparse point clouds without explicit mesh geometry, direct path tracing is non-trivial. Inspired by deferred shading, we follow GS-IR [95] and reconstruct view-space geometry from depth and normal maps. Path tracing is then performed over this reconstructed surface using a tile-based acceleration structure to improve ray marching efficiency.

Given the depth $d(u, v)$ and normal maps, we recover the 3D position $\mathbf{x} = (x, y, z)^T$ for

each screen-space pixel (u, v) by inverse projection:

$$\mathbf{x} = z \cdot (u - c_x, v - c_y, 1)^T,$$

where (c_x, c_y) is the principal point of the camera.

To compute ambient occlusion, we integrate the visibility function $V(\omega)$ over the hemisphere:

$$O(\mathbf{x}) = 1 - \frac{1}{\pi} \int_{\Omega} V(\omega) \, (\mathbf{n} \cdot \omega) \, d\omega, \tag{6.19}$$

where ω is the sampled incoming direction. For each ray from $\mathbf{x}$ in direction ω, we check whether it is occluded by comparing the depth z of the sampled point $\mathbf{x} + t\omega$ with the expected depth z_d at the projected screen position. The visibility is defined as:

$$V(\omega) = \begin{cases} 1 & \text{if } z_d < z < z_d + \delta, \\ 0 & \text{otherwise,} \end{cases} \tag{6.20}$$

This occlusion test is performed adaptively in screen space based on perspective projection (see Figure 6.3(a)).

For indirect lighting, we assume that if a ray r_i intersects another surface point $\hat{\mathbf{x}}$, the incoming radiance can be approximated by the direct lighting value L_{dir} at that point:

$$L_i(\omega_i, \mathbf{x}) = V(\omega_i, x) \, L_{dir}(\omega_i, x), \tag{6.21}$$

which yields the indirect component integrated over the hemisphere:

$$L_{ind} = (1 - m) \frac{a}{\pi} \int_{\Omega} L_i(\omega_i, \mathbf{x}) \, (\mathbf{n} \cdot \omega_i) \, d\omega_i. \tag{6.22}$$

Outgoing Light Modeling The final outgoing radiance combines direct and indirect illumination with a specular term:

$$L_o(\omega_o, \mathbf{x}) = L_{dir} + L_{ind}. \tag{6.23}$$

To supervise the learning of material and illumination components, we define the following decomposition loss:

$$\mathcal{L}_d = \mathcal{L}_1 + \lambda_M \mathcal{L}_{\text{TV}_{\text{mat}}} + \lambda_E \mathcal{L}_{\text{TV}_{\text{light}}} \tag{6.24}$$

where $\mathcal{L}_1$ is the pixel-wise reconstruction loss (Equation 6.25) between the rendered image and the ground truth. The total variation regularization $\mathcal{L}_{\text{TV}_{\text{mat}}}$ (Equation 6.26) and $\mathcal{L}_{\text{TV}_{\text{light}}}$ (Equation 6.27) enforce spatial smoothness on the predicted material and lighting

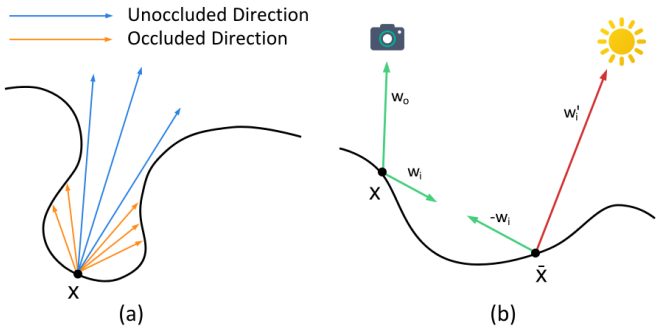

Figure 6.4: (a) Surface points with less visibility receive weaker indirect illumination. (b) Path tracing is performed on G-buffer-reconstructed geometry to evaluate visibility and indirect radiance.

maps, reducing noisy or fragmented estimates. The weights λ_M and λ_E balance these regularizers against the reconstruction loss and stabilize training.

$$\mathcal{L}_1 = \sum_i ||I_{render}(i) - I_{gt}(i)||_1 \tag{6.25}$$

$$\mathcal{L}_{TV}^{mat} = \sum_{x,y} ||M(x+1,y) - M(x,y)||_1 + ||M(x,y+1) - M(x,y)||_1 \tag{6.26}$$

$$\mathcal{L}_{TV}^{light} = \sum_{x,y} ||L(x+1,y) - L(x,y)||_1 + ||L(x,y+1) - L(x,y)||_1 \tag{6.27}$$

To support efficient estimation of indirect lighting, we render a cubemap from each surface point by capturing the scene from six orthogonal directions. This cubemap encodes occlusion and incoming radiance information from all directions, allowing visibility-aware light gathering during path tracing. Figure 6.4(b) shows the occlusion-aware geometry, and Figure 6.3(b) illustrates cubemap-based integration. This process corresponds to the integral in Equation 6.22 and enables efficient evaluation of L_{ind} in Equation 6.23.

6.5 Experiments

6.5.1 Experiment Settings

We adopt a self-supervised training strategy in this work, leveraging a pretrained state-of-the-art generative Gaussian splatting model named DiffSplat [96]. We divide our experiments into two parts: 3D generation and material decomposition.

For the generation experiment, we conduct both text-conditioned and image-conditioned generation. In the text-conditioned setting, we use captions from a subset of Cap3D [104] to generate Gaussian splats, decompose the materials and illumination of the generated

3D scenes, and then reconstruct the decomposed scenes to evaluate generation quality. We use 300 text prompts from T3Bench [47], covering single objects, single objects with surroundings, and multiple-object scenes. We report CLIP similarity [122], CLIP R-Precision [118] with ViT-B/32, and ImageReward [165] to measure prompt alignment and overall perceptual preference.

For the image-conditioned generation task, we randomly select 300 objects from the unseen GSO [26] dataset and render them to serve as ground-truth images. Our model reconstructs and decomposes the 3D scene of the input image, including geometry, material, and illumination, and then re-renders the decomposed scene for evaluation. We compare the rendered images from reconstructed or generated 3D content using PSNR, SSIM, and LPIPS [186]. All metrics are averaged across viewpoints to reflect 3D-aware quality rather than a single canonical view. For the material decomposition experiment, we follow the evaluation logic of MaterialFusion [98]: we relight the decomposed objects under novel illumination and compare the results with ground truth. For synthetic objects, the ground truth is obtained by rendering under unseen illuminations; for real objects, the references are captured in novel environments with recorded lighting. We again report PSNR, SSIM, and LPIPS across all relighting tests.

Table 6.1: Quantitative evaluation of text-conditioned scene generation

		DiffGSPBR(Ours)	DiffSplat	GVGEN	LN3Diff	DIRECT-3D	3DTopia	LGM	GRM
Single Object	↑ CLIP Sim.%	30.52	**30.95**	23.66	24.36	24.80	25.55	29.96	28.19
	↑ CLIP R-Prec.%	78.00	**81.00**	23.25	27.25	30.75	34.50	78.00	64.75
	↑ ImageReward	-0.628	**-0.491**	-2.156	-2.008	-2.005	-1.998	-0.720	-1.337
Single Object w/ Sur.	↑ CLIP Sim.%	**30.37**	30.20	22.65	22.75	23.05	24.31	27.79	26.24
	↑ CLIP R-Prec.%	**83.50**	80.75	26.75	22.00	25.75	39.00	55.00	51.25
	↑ ImageReward	-1.184	**-0.674**	-2.251	-2.244	-2.191	-2.230	-1.772	-1.869
Multiple Objects	↑ CLIP Sim.%	27.34	**29.46**	21.48	21.65	21.89	22.88	27.07	24.33
	↑ CLIP R-Prec.%	58.50	**69.50**	8.00	8.75	7.75	16.50	51.00	26.50
	↑ ImageReward	-1.645	**-0.849**	-2.272	-2.267	-2.249	-2.225	-1.731	-2.116

6.5.2 Text-conditioned Scene Generation

Baselines We compare our model with state-of-the-art text-to-3D generation methods including the Gaussian-based DiffSplat [96], GVGEN [46], the triplane-based LN3Diff [83], DIRECT-3D [100], 3DTopia [54], and the reconstruction-based methods LGM [147] and GRM [166] coupled with an open-source text-conditioned multi-view diffusion model [140]. These baselines are appropriate because they cover both native 3D generation and reconstruction-based pipelines, including the strongest recent Gaussian-splatting comparator.

Comparisons Table 6.1 and Figure 6.5 present the quantitative and qualitative comparison of our model against the baselines. The results show that our model remains competitive with strong text-conditioned 3D generators, indicating that the decomposition pipeline

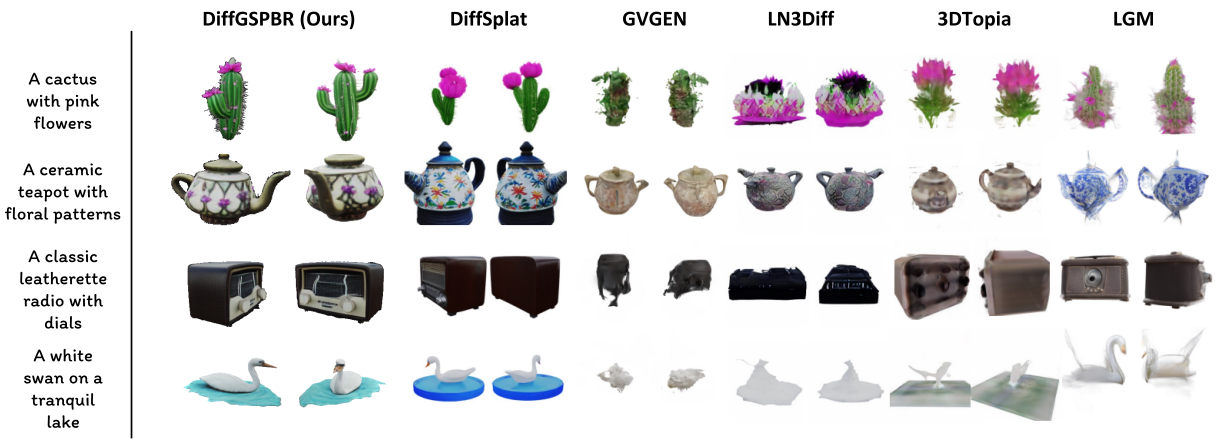

Figure 6.5: Results of text-conditioned scene generation

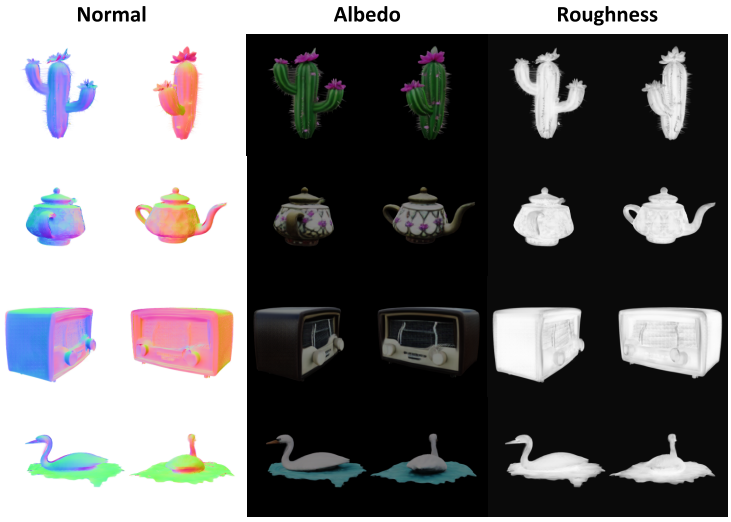

Figure 6.6: Material estimation from text condition.

does not substantially degrade generation quality while preserving editability. Figure 6.5 further shows that our method maintains prompt alignment and visual quality close to recent text-conditioned 3D generation methods [96]. We also show the material-estimation results in Figure 6.6, where the model produces coherent normal, albedo, and roughness maps for scenes represented by Gaussian splatting.

6.5.3 Image-conditioned Scene Generation

Baselines We compare against two up-to-date native 3D models that support image-conditioned generation: the concurrent work 3DTopia-XL [17] and LN3Diff [83]. We also evaluate six advanced reconstruction-based methods for single-image-conditioned generation, including four Gaussian-splatting-based methods, DiffSplat [96], LGM [147], GRM [166], and LaRa [13], as well as two FlexiCube-based [136] methods, CRM [158] and InstantMesh [164]. These baselines are appropriate because they span native 3D generation

and reconstruction-based pipelines, allowing us to compare against the main families of image-conditioned 3D synthesis methods. Image generative models for these reconstruction methods are selected following their original implementations.

Table 6.2: Quantitative evaluation of image-conditioned scene generation

	Ours	DiffSplat	3DTopia-XL	LN3Diff	LGM	GRM	LaRa	CRM	InstantMesh
↑ PSNR	22.36	**22.91**	17.27	16.67	18.25	19.65	18.87	18.56	19.14
↑ SSIM	**0.914**	0.892	0.840	0.831	0.841	0.869	0.852	0.855	0.876
↑ LPIPS	**0.081**	0.107	0.1756	0.177	0.1665	0.141	0.2020	0.149	0.128

Comparisons Single-image-conditioned generation performance on the GSO dataset is assessed in Table 6.2, and qualitative results on in-the-wild images are presented in Figure 6.8. DiffGSPBR generates accurate 3D content aligned with the input images while maintaining strong geometric fidelity. The results are close to the state-of-the-art DiffSplat [96] and substantially better than the remaining baseline models. Figure 6.8 illustrates the material decomposition results of image-conditioned synthesis; the predicted materials capture both the shape and the appearance cues of the input image, supporting the effectiveness of the decomposition stage.

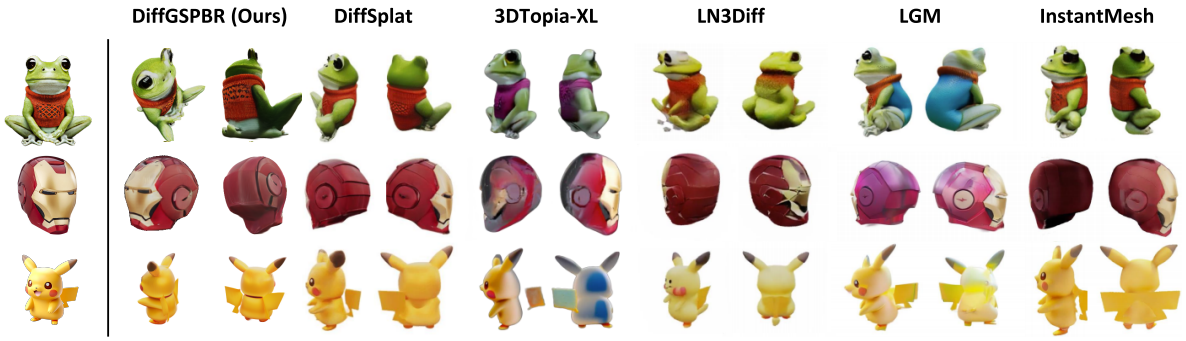

Figure 6.7: Results of image-conditioned scene generation

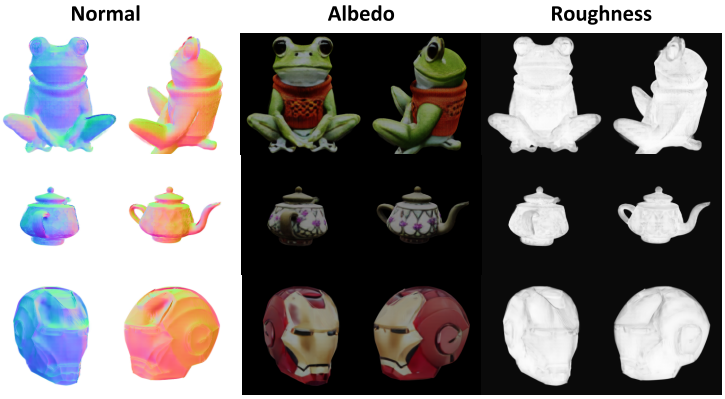

Figure 6.8: Material estimation from image condition.

6.5.4 Application: Relighting

We conduct a relighting experiment to evaluate the effectiveness of the material estimation stage. Figure 6.9 illustrates the relighting results obtained by rendering the estimated material maps under a new environment light. The decomposed materials produced by our model respond consistently to the new illumination, reflecting the incident light while producing plausible hue changes and shadows. Consequently, the model captures light-surface interaction more faithfully, and the path-tracing-based Gaussian deferred renderer handles shadowed and occluded regions effectively.

Figure 6.9: Relighting experiment for decomposed 3D content.

6.6 Conclusion

In this chapter, we presented DiffGSPBR, a diffusion-based Gaussian splatting framework that extends the thesis from relightable 2D generation to relightable and view-controllable 3D generation. By combining generative 3D Gaussian splats with material and light decomposition, the method preserves competitive generation quality while making the resulting scene editable in ways that conventional text-to-3D pipelines do not directly support.

The method also remains limited in several respects. The decomposition is self-supervised and can therefore drift when the generated geometry is weak or when materials are highly specular. In addition, the path-tracing-based deferred renderer increases computational cost relative to purely appearance-based 3D generators, and the current experiments focus on object-centric scenes rather than full dynamic environments. Even with these limitations, the chapter completes the thesis trajectory from latent control, to 2D rendering-aware generation, to editable 3D scene synthesis.

Chapter 7

Conclusions and Future Work

7.1 Contributions and Conclusions

This thesis investigated how explicit rendering structure can be used to improve the controllability of generative models. The main argument is that, for many graphics-oriented tasks, it is not enough for a model to produce an image that simply looks realistic. If the result is meant to be edited after it is generated, then key factors such as pose, material, illumination, and viewpoint cannot all be mixed together inside a black-box generator. They need to be represented in a way that can still be accessed and modified afterwards.

Rather than treating image formation as an implicit by-product of sampling, this thesis developed a sequence of methods that progressively separate content generation from rendering. This thesis did not begin by assuming that a single end-to-end architecture could solve all aspects of controllable generation at once. It first examined how controllability can be studied inside an existing latent generator, then isolated relighting as a rendering problem in its own right, and only after that integrated rendering-aware intermediate representations into 2D and 3D generative pipelines. In this way, this thesis moved from analysis of latent controllability, to learnable shading and decomposed scene generation.

The resulting contributions therefore address three connected levels of the problem: controllability inside a latent generator, learned shading for real-world relighting, and the integration of rendering-aware intermediate representations into 2D and 3D generative pipelines. Taken together, these contributions support the broader conclusion that explicit rendering structure is not merely an implementation detail, but a useful organising principle for controllable generative modeling.

7.1.1 Contributions

The main contributions of this thesis can be concluded as follows:

- **Chapter 3.** We analysed controllability inside StyleGAN and introduced *generative fields* as a way to characterise the spatial extent of channel-wise control. This contribution is important because it shifts the discussion of latent control away from purely empirical design and toward an explicit structural interpretation of how different generator layers affect different image regions and semantic scales. Based on this analysis, we developed a reference-guided editing framework for pose and expression manipulation that preserves identity while transferring selected facial attributes. The contribution of this chapter is therefore twofold: it provides a conceptual tool for analysing latent controllability, and it demonstrates that such analysis can be translated into a practical editing method.

- **Chapter 4.** We proposed a *physics-based neural deferred shading* pipeline for controllable relighting of real-world images. Rather than treating relighting as a purely generative image transformation, this chapter formulates it as a learnable rendering problem conditioned on estimated material, geometry, and illumination cues. The method combines a neural shader with an additional shadow-estimation stage so that relighting remains tied to explicit scene attributes rather than being left entirely implicit inside a generative model. To support this study, we introduced *FFHQ256-PBR*, a facial dataset with estimated material, geometry, ambient occlusion, illumination, and camera attributes. The contribution of this chapter is therefore not only a relighting model, but also a rendering-based formulation of illumination control for real-world portrait data.

- **Chapter 5.** We introduced *ShadingFusion*, a rendering-aware text-to-image pipeline that predicts decomposed material representations and then renders them with a neural shader. This contribution addresses a key limitation of standard text-to-image diffusion systems: they usually commit directly to final RGB appearance, which entangles content, material, and illumination in a single output. Our method instead shifts the generative target toward editable intermediate representations, using a modified multi-head VAE, prompt enrichment through an MLLM, and a rendering stage that reconstructs the final image under controllable lighting. In this sense, the contribution of Chapter 5 is to show that diffusion-based synthesis can preserve photorealism while becoming more editable, provided that generation and rendering can be explicitly separated.

- **Chapter 6.** We proposed *DiffGSPBR*, which extends the same decomposition principle to 3D Gaussian splatting and supports editable 3D generation with controllable material, illumination, and viewpoint. This chapter is significant because it moves beyond image-plane relighting and treats decomposition as part of the 3D scene representation itself. The method combines an off-the-shelf generative Gaussian

backbone with material and illumination estimation heads and a deferred rendering loop for self-supervised optimisation. As a result, geometry, material, and lighting can be manipulated after generation in a way that is not naturally available in conventional 3D generative pipelines. The contribution of this chapter is therefore to show that rendering-aware controllability can be extended from 2D portrait synthesis to editable 3D scene generation.

7.1.2 Conclusions

The main conclusions of this thesis are as follows:

Controllability benefits from explicit intermediate structure. Chapter 3 shows that controllability inside a generator is easier to analyse when the roles of different latent channels and layers are made explicit. In practice, this means that control becomes more reliable when it is tied to some interpretable internal structure rather than treated as a purely empirical editing problem. Although this does not by itself solve rendering-aware editing, it provides the representational foundation for later chapters by showing that generation can be made more controllable when intermediate structure is studied directly.

Illumination can be treated as a learnable rendering problem. Chapter 4 shows that a neural deferred shading model can learn a useful approximation of the rendering process from estimated material, geometry, and lighting cues, making relighting possible for real-world imagery without hand-crafted shading models. More importantly, it shows that relighting does not have to be handled only as a stochastic image-generation problem. When explicit scene attributes are available, illumination control can instead be organised as a constrained rendering process, which gives stronger interpretability and more predictable behaviour.

Separating generation from rendering improves editability in 2D synthesis. Chapter 5 demonstrates that a text-to-image pipeline can be made more controllable by generating decomposed intermediate representations and rendering them explicitly, rather than predicting final RGB images alone. This conclusion is important because it shows that diffusion-based image synthesis does not have to choose between realism and editability. By shifting the generative target from final RGB appearance to a more structured intermediate representation, the system can preserve strong image quality while still supporting explicit control over lighting and material behaviour after generation.

The same principle extends to 3D generation. Chapter 6 shows that rendering-aware decomposition is also useful in a 3D Gaussian representation, where material, illumination, and viewpoint can be manipulated more explicitly after generation. This suggests that the value of explicit rendering structure is not limited to image-space relighting or portrait

synthesis. The same idea can also support editable scene-level generation in 3D, where the need for controllable viewpoint and lighting is even more fundamental.

7.1.3 Response to Thesis Research Questions

Taken together, the results of the thesis provide a coherent response to the research questions posed in the introduction. These questions were not answered by a single unified architecture. Instead, they were addressed progressively, with each later chapter building on the representational or rendering insights established earlier. The responses can therefore be summarised as follows:

- **RQ1.** Chapter 3 shows that fine-grained control can be improved by analysing the internal structure of a generator and by applying channel-wise control at appropriate spatial scales. This provides a useful but still limited form of controllability inside an implicit image generator. The main answer is that latent or intermediate representations are more effective for control when they are studied in terms of their internal roles and spatial influence, rather than treated as a single control signal.

- **RQ2.** Chapter 4 shows that the shading process can be learned as a neural deferred rendering problem using estimated material, geometry, and illumination cues. This makes illumination control more explicit and provides the rendering component required for later generative integration. More specifically, the thesis shows that relighting can be organised as a rendering process conditioned on explicit scene attributes, rather than only as a stochastic image-synthesis problem, which leads to stronger interpretability and more reproducible control.

- **RQ3.** Chapters 5 and 6 show that rendering-aware intermediate representations can be integrated into both 2D and 3D generative pipelines. In 2D, this enables text-to-image synthesis with editable material and illumination. In 3D, it enables relightable Gaussian-based generation with explicit viewpoint control. The broader answer is that explicit rendering structure can be inserted into modern generative systems without giving up their expressive power, and that doing so makes the generated outputs substantially more useful for post-generation editing.

Overall, this thesis concludes that explicit rendering structure is a practical way to improve the editability of generative models. The main outcome is not a single final architecture, but a staged demonstration that controllable generation becomes more feasible when image-formation variables are represented explicitly rather than absorbed into an opaque RGB generator. In this sense, the contribution of the thesis is both methodological and conceptual: it proposes concrete methods for relighting and decomposed generation, and at the same time argues for a broader way of thinking about controllable generative

modeling, in which rendering is treated as a first-class component rather than a hidden by-product of image synthesis.

7.2 Limitations

The methods proposed in this thesis also have several limitations.

Chapter 3 limitations. The StyleGAN-based editing framework is limited to the representational capacity and domain of the underlying generator. It provides fine-grained control over selected facial attributes, but does not expose physically meaningful variables such as illumination or material.

Chapter 4 limitations. The neural deferred shading model relies on estimated material and lighting cues rather than fully ground-truth physical annotations. Its performance is therefore affected by upstream estimation quality and by the domain gap between synthetic assets and real-world imagery.

Chapter 5 limitations. ShadingFusion improves controllability by predicting decomposed intermediate representations, but the setting remains focused on human faces and inherits the limitations of estimated PBR supervision and text-description quality.

Chapter 6 limitations. DiffGSPBR extends controllable generation to 3D, but it remains restricted by the expressive range of the chosen Gaussian representation, the quality of learned decomposition, and the computational cost of rendering-aware training and evaluation.

More generally, this thesis does not claim that all controllability problems should be solved through explicit rendering decomposition. Instead, it shows that this strategy is useful for a class of tasks where material, illumination, and viewpoint must remain editable within generation.

7.3 Directions for Future Work

Building on these findings, several promising avenues remain and can be discovered in future work:

Domain Adaptation for Real-World Material Distributions. While our neural deferred shaders generalize to HDRI-based relighting, they do not explicitly model the domain shift between synthetic PBR maps and estimated real-world materials. Incorporating unsupervised domain adaptation or self-supervised fine-tuning could further improve performance on in-the-wild imagery.

Extension to Dynamic and Complex Scenes. Both ShadingFusion and DiffGSPBR

currently target single-object or portrait scenarios. Future work should extend these methods to handle multi-object compositions, dynamic lighting, and non-rigid scene elements (e.g., cloth, fluids).

Real-Time Inference and Optimization. Although our deferred renderers operate efficiently at inference, real-time performance on high-resolution inputs remains a challenge. Further optimization of network architectures and hybrid approximations, such as learned importance sampling, could enable interactive applications in gaming and virtual production.

Material and Illumination Editing Interfaces. Developing user-friendly tools (e.g., slider-based UIs or language-driven editing prompts) built upon our pipelines would empower artists and designers to intuitively manipulate material and lighting attributes in generated content.

Integration with Inverse Rendering and Scene Understanding. By coupling our framework with inverse rendering approaches that estimate geometry and lighting from photographs, one could enable bidirectional workflows: edit existing scenes, or generate new content with specified physical properties.

7.4 Closing Remarks

This work bridges generative modeling and classical computer graphics, showing that neural deferred shading can unlock precise illumination control in both image and 3D synthesis. As generative models continue to advance, integrating physics-based rendering components will be essential for creating assets that are not only photorealistic but also fully editable and integrable into traditional content-creation pipelines. We believe the methodologies presented here will serve as a foundation for future research at the intersection of machine learning and rendering, driving new capabilities in visual effects, virtual production, and beyond.

Appendix A

Generative Fields

A.1 Generative fields size for all convolution layers

Table A.1 provides the generative field size of each convolution layer in style space $\mathcal{S}$, we calculate them by using the generative fields formula defined in section 3.2 of the paper. All convolution layers are indexed from `conv0` to `conv12` for the pre-trained StyleGAN2 generator with 256×256 resolution, we refer $\mathcal{S}$ layer index from [162], items that have same color in the table belong to the same generator block of StyleGAN2.

input resolution	$\mathcal{S}$ layer index	conv layer index	generative fields	# channels
4×4	s0	conv0	506	512
8×8	s2	conv1	379	512
8×8	s3	conv2	251	512
16×16	s5	conv3	187	512
16×16	s6	conv4	123	512
32×32	s8	conv5	91	512
32×32	s9	conv6	59	512
64×64	s11	conv7	43	512
64×64	s12	conv8	27	256
128×128	s14	conv9	19	256
128×128	s15	conv10	11	128
256×256	s17	conv11	7	128
256×256	s18	conv12	3	64

Table A.1: Generative fields size of each convolution layer in StyleGAN2

A.2 Comparison of image editing results

We provide a comparison for the image editing results with different functional generative field size (GFs), the attribute images are randomly sampled from real world dataset FFHQ256, identity images are randomly generated from a pre-trained StyleGAN2 generator.

As we mentioned in the paper (Sec. 4.2.2), the average face size of dataset is 141.68 pixels, Figure A.1 illustrates the comparison of feature editing results where smallest functional generative field sizes are below and above the average face size, it could obviously find that the model lose the expression feature control when the minimum generative field size is higher than the average face size.

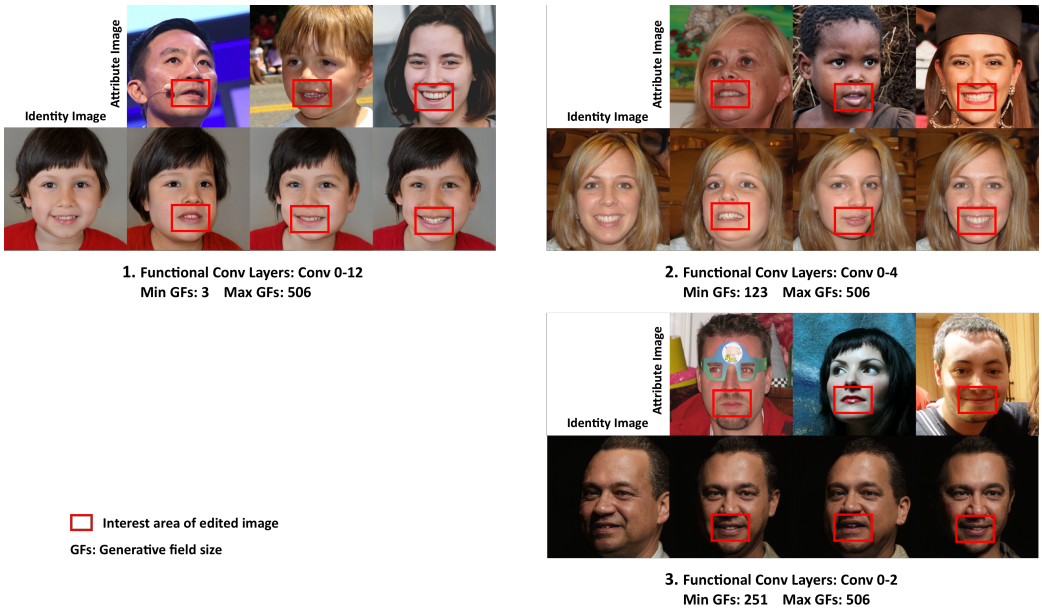

Figure A.1: Image editing results comparison. `Image1` is the default setting which enables all GFs; the minimum GFs of `image2` is below the average face size; the minimum GFs of `image3` is above the average face size.

We also compare image editing results which only use smaller generative fields. The largest generative field size of `image1` is slightly higher than the average face size which can reserve a little pose editing capacity with few torn artifact; for `image2`, the largest generative field size is lower than the average face size, it doesn't has enough influencing area to edit the whole head feature, then losing the head pose control.

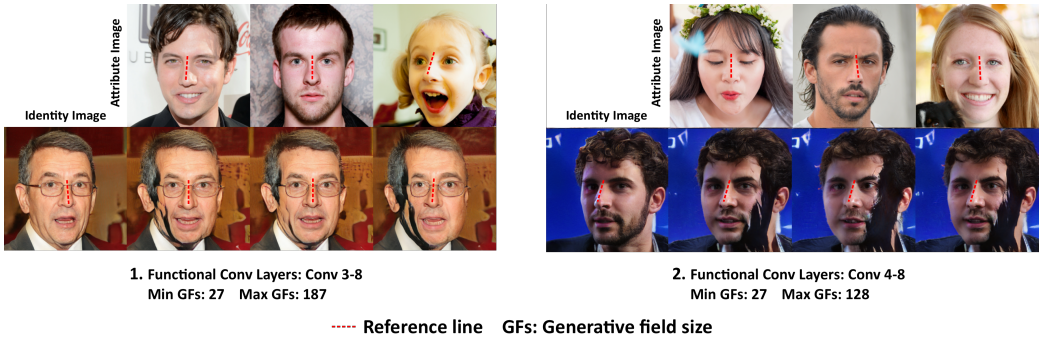

Figure A.2: Image editing results comparison using limited generative field size.

We find if only enabling very small generative field sizes the model would force to

match the head pose and expression by downgrading the generated image quality (See Figure A.3), which could consist of many torn areas or pulled facial features. We explain it through it could cheat the pre-trained landmark detector to produce a matching result (See the overlap of detected landmark in Figure A.3).

Figure A.3: Image editing results with generative field size from 7 to 59.

Appendix B

ShadingFusion

B.1 Preset Questions for Facial Information Acquisition

Training information acquisition We use the multi-modality large language model (MLLM), DeepSeek-VL2-32B, to perform the Visual Question Answering (VQA) task, we design several questions to ask the MLLM to retrieve the required textual information from the input portrait image, the specific questions are as follow:

System prompt:

You are a vision-language assistant. When asked to summarize a face, output a single, coherent paragraph less than 80 words. The paragraph must start with a clause formatted like: A portrait photo of a <**young** / middle-aged / **older** > <**man** / **woman** > in the age range of <low>-<high>.

User Questions/Instructions:

(1) Please identify the person's gender and approximate age. <If not provided, choose one randomly>

(2) Describe the overall face shape, facial contour (jawline, cheekbones, forehead) and the accessories (if have). <If the description is not provided, generate a description of human face shape, facial contour and accessories followed by the instructions.>

(3) How are the facial features proportioned? Comment on spacing between eyes, nose, mouth and overall symmetry. <If not provided, generate a description of human facial feature followed by instructions.>

(4) Describe the eyes and eyebrows: eye shape/size, eyelid type, eye-corner tilt, iris colour; eyebrow shape and density. <If not provided, generate a description of human eyes and eyebrows followed by instructions.>

(5) Describe the nose: bridge height, length, width of the nostrils, and tip shape. <If not provided, generate a description of human nose followed by instructions.>

(6) Describe the mouth and chin: lip thickness/shape, mouth-corner tilt, chin type and

jaw definition. <If not provided, generate a description of human mouth and chin followed by instructions.>
(7) Describe the skin colour and overall surface quality: Skin tone — name the overall shade (e.g., pale ivory, light olive, medium tan, deep brown) and note the undertone (cool/pink, neutral, or warm/golden). Overall finish — say whether the skin looks matte, dewy, oily, or dry, without listing fine details. <If not provided, generate a description of human skin color and quality followed by instructions.>
(8) In 1–2 sentences, describe visible skin-texture details: Wrinkles — depth and location; Pores / pits — enlarged or normal, location; Discolorations — freckles, moles, scars, redness; size and location. <If not provided, generate a description of human skin texture details followed by instructions.>
(9) Describe hair and facial hair: hair colour, length, texture, hairline; presence and style of beard/moustache. <If not provided, generate a description of human hair and beard followed by instructions.>
(10) Combine all the details you have given into a single, coherent paragraph (less than 80 words) that fully describes the person's appearance and the accessories (if have), mentioning gender, age range, geometry features, and texture details. Do **not** repeat any phrase. Each attribute only once. Keep the language natural.

The MLLM answers all those questions and compiles the final result into a detailed description of facial attributes, as shown in Figure B.1. The colored text labels the predefined options in the system prompt. All collected answers are stored in a JSON file for the following training procedures.

Inference prompt augmentation During inference, we apply the similar process for the user prompt augmentation. The MLLM rephrases each prompt using the full set of questions and instructions from our training information acquisition procedure then appends an additional cyan-highlighted sentence to preserve the prompt's structural integrity for diffusion model training.

B.2 Normalization Method

Since the depth values span a wide range, we first establish a clipping space bounded by a near plane Z_{near} and a far plane Z_{far}, and then map the depth data into that space by clipping any values outside those bounds. We set the near and far planes to 1 and 800 respectively, and the normalization is defined as follows:

$$D_{norm} = \frac{D - Z_{near}}{Z_{far} - Z_{near}}$$

B.3 Loss Function of Mask Prediction

We combine several commonly used segmentation losses, including Binary Cross Entropy (BCE) loss [35], DICE loss [131], and boundary loss [76], for mask-prediction training. These losses are defined as follows.

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[M_i \log(\hat{M}_i) + (1 - M_i)log(1 - \hat{M}_i)]$$

$$\mathcal{L}_{DICE} = 1 - \frac{2\sum M_i \cdot \hat{M}_i}{\sum M_i + \sum \hat{M}_i}$$

$$\mathcal{L}_{Boundary} = \int_{\Omega} |\nabla P| dx, \;\; P = Softmax(\hat{M})$$

$$\mathcal{L}_{Mask} = \mathcal{L}_{BCE} + \mathcal{L}_{DICE} + \mathcal{L}_{Boundary}$$

B.4 Importance Sampling

In image-based lighting (IBL), importance sampling [153] is commonly used to estimate the integral of incident radiance from an HDR environment map, as given by the following equation.

$$L_o(x, \omega_o) \approx \frac{1}{N}\sum_{i=1}^{N} \frac{L_i(\omega_i) f_r(x, \omega_i, \omega_o)|cos\theta_i|}{p(\omega_i)}, \;\; \omega_i \sim p(\omega)$$

where $L_o(x, \omega_o)$ is the outbound radiance, $L(\omega_i)$ is the incident radiance along ω_i direction, $f_r(x, \omega_i, \omega_o)$ is the bidirectional reflectance function (BRDF), $p(\omega_i)$ is the probability density function (PDF) used by importance sampling.

For a shading point, importance sampling chooses incident-light directions according to the probability density induced by the environment map, allocating more samples to high-intensity regions and fewer to low-intensity ones. This strategy significantly reduces the variance of the estimated incident radiance, thereby lowering rendered noise compared with uniform sampling.

Decription Example 01:

A portrait photo of a young woman with long flowing straight silky blonde hair past her shoulders, fair smooth radiant complexion, well-proportioned delicate jawline, prominent cheekbones, slightly pointed chin, large almond-shaped blue eyes with thick lashes and gently tilted eyelids, well-groomed arched eyebrows, narrow nose with defined bridge and small nostrils, full softly upturned lips, strikingly symmetrical features, subtle minimal accessories or makeup, natural beauty, warm inviting expression, youthful glow, ethereal aura, untouched elegance, refined proportions, graceful poise, timeless allure.

Input Image

Decription Example 02:

A portrait photo of a older man with a slightly oval face, pronounced cheekbones, defined jawline, broad forehead with visible wrinkles, blue almond-shaped eyes under thick arched eyebrows, straight nose with moderate bridge and flared nostrils, full lips with a subtle upward curve, neat moustache and goatee, light brown hair streaked with gray combed back, fair skin showing fine lines and natural texture variations, smooth healthy sheen, mature dignified presence reflecting wisdom, refined character, subtle elegance, timeless aura, serene demeanor.

Input Image

Prompt Example 01:

User prompt: A portrait photo of a middle-aged man with long hair.

Augmented prompt: A portrait photo of a middle-aged man with long straight brown hair past his shoulders, neatly trimmed beard, prominent forehead, high cheekbones, defined jawline, almond-shaped eyes with glasses and dark irises under thick slightly arched eyebrows, medium nose with gently sloping bridge and moderate nostrils, average-sized mouth with neutral expression and full lips, smooth skin with minor wrinkles at eyes and forehead, no significant blemishes, balanced symmetrical features, calm composed demeanor, refined mature elegance, natural understated timeless serene presence. Looking at the screen under the wild light.

Generated Image

Prompt Example 02:

User prompt: A portrait photo of a middle-aged woman with long wavy hair.

Augmented prompt: A portrait photo middle-aged woman with long wavy brown hair cascading over her shoulders, smooth lustrous texture, a heart-shaped face with prominent cheekbones, almond-shaped expressive eyes with thick lashes and slight upward tilt, neatly shaped dense arched eyebrows, straight nose with well-defined bridge and narrow nostrils, full naturally pink lips curved in a subtle smile, smooth warm-toned skin with minimal fine lines around mouth and forehead, absence of moles or scars, balanced symmetrical features, mature elegance, serene poise. Looking at the screen under the indoor light.

Generated Image

Figure B.1: Facial information examples

Bibliography

[1] Titas Anciukevičius, Fabian Manhardt, Federico Tombari, and Paul Henderson. Denoising diffusion via image-based rendering. In *The Twelfth International Conference on Learning Representations*, 2024.

[2] Titas Anciukevičius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J Mitra, and Paul Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12608–12618, 2023.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.

[4] Michael Ashikmin, Simon Premože, and Peter Shirley. A microfacet-based brdf generator. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 65–74, 2000.

[5] Harry Barrow, J Tenenbaum, A Hanson, and E Riseman. Recovering intrinsic scene characteristics. *Comput. vis. syst*, 2(3-26):2, 1978.

[6] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. GAN dissection: Visualizing and understanding generative adversarial networks. In *7th International Conference on Learning Representations, 2019*.

[7] Song Bi, Yusuke Tachibana, Roey Mechrez, Matthew Fisher, Dani Lischinski, and Daniel Cohen-Or. Neural-ray-bridge: Bridging neural rendering and simulated ray tracing. In *SIGGRAPH Asia*, 2021.

[8] Mikolaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[9] James F. Blinn. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '77, pages 192–198, July 1977.

[10] Mark Boss et al. Neural-PIL: Neural pre-integrated lighting for reflectance decomposition. In A. Beygelzimer et al., editors, *Advances in Neural Information Processing Systems*, 2021.

[11] Ziang Cao, Fangzhou Hong, Tong Wu, Liang Pan, and Ziwei Liu. Large-vocabulary 3d diffusion model with transformer. *arXiv preprint arXiv:2309.07920*, 2023.

[12] Nic Carr, Jim Hall, and Mark Green. Advances in real-time shading. In *SIGGRAPH Courses*, 2002.

[13] Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *European Conference on Computer Vision*, pages 338–355. Springer, 2024.

[14] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2416–2425, 2023.

[15] Hongze Chen, Zehong Lin, and Jun Zhang. Gi-gs: Global illumination decomposition on gaussian splatting for inverse rendering. *arXiv preprint arXiv:2410.02619*, 2024.

[16] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[17] Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, et al. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26576–26586, 2025.

[18] Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21401–21412, 2024.

[19] Adam Coates and Andrew Ng. Selecting receptive fields in deep networks. *Advances in neural information processing systems*, 24, 2011.

[20] Robert L Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1(1):7–24, 1982.

[21] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs. In *SIGGRAPH*, pages 11–20, 1996.

[22] Christopher DeCoro and Christopher Everitt. Deferred shading in dynamic scenes. In *Game Developers Conference*, 2005.

[23] Michael Deering et al. The triangle processor and normal vector shader: a VLSI system for high performance graphics. *SIGGRAPH Comput. Graph.*, pages 21–30, August 1988.

[24] Christian Dick, Jens Schneider, and Rüdiger Westermann. Efficient geometry compression for gpu-based decoding in realtime terrain rendering. In *Computer Graphics Forum*, volume 28, pages 67–83. Wiley Online Library, 2009.

[25] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *ICLR*, 2017.

[26] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022.

[27] Patrick Esser, Robin Rombach, and Andreas Blattmann. Finetuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2408.09275*, 2024.

[28] Ali Farhadi and Joseph Redmon. Yolov3: An incremental improvement. In *Computer vision and pattern recognition*, volume 1804, pages 1–6. Springer Berlin/Heidelberg, Germany, 2018.

[29] Yao Feng et al. Learning an animatable detailed 3D face model from in-the-wild images. *ACM Trans. Graph.*, pages 1–13, August 2021.

[30] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[31] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. In *PAMI*, volume 32, pages 1362–1376, 2010.

[32] Aviv Gabbay and Yedid Hoshen. Demystifying inter-class disentanglement. In *8th International Conference on Learning Representations (ICLR)*, 2020.

[33] Jian Gao, Chun Gu, Youtian Lin, Zhihao Li, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussians: Realistic point cloud relighting with brdf

decomposition and ray tracing. In *European Conference on Computer Vision*, pages 73–89. Springer, 2024.

[34] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[35] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[36] Ian Goodfellow et al. Generative adversarial networks. In *NeurIPS*, pages 2672–2680, 2014.

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

[38] Gianluca Guarini, Maurizio Rossi, et al. Pbr material: a comparison between rendering for accurate material color reproduction. a case study. *Color and Colorimetry Multidisciplinary Contributions*, page 38, 2024.

[39] Yuan-Chen Guo et al. NeRFReN: Neural Radiance Fields with Reflections. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18388–18397, June 2022.

[40] Saeed Hadadan et al. Neural radiosity. *ACM Trans. Graph.*, pages 1–11, December 2021.

[41] Junlin Han, Filippos Kokkinos, and Philip Torr. Vfusion3d: Learning scalable 3d generative models from video diffusion models. In *European Conference on Computer Vision*, pages 333–350. Springer, 2024.

[42] Yuxuan Han, Zhibo Wang, and Feng Xu. Total relighting: learning to relight portraits for background replacement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 8598–8608. IEEE, 2023.

[43] Hanting He, Yifan Zhang, Zhen Liu, Qi Chen, Aiwu Zhou, and Ming Song. Lightit: Photorealistic environment-aware portrait relighting from a single image. *arXiv preprint arXiv:2404.14449*, 2024.

[44] Kai He, Ruofan Liang, Jacob Munkberg, Jon Hasselgren, Nandita Vijaykumar, Alexander Keller, Sanja Fidler, Igor Gilitschenski, Zan Gojcic, and Zian Wang. Unirelight: Learning joint decomposition and synthesis for video relighting. *arXiv preprint arXiv:2506.15673*, 2025.

[45] Mingming He, Pascal Clausen, Ahmet Levent Tasel, Li Ma, Oliver Pilarski, Wenqi Xian, Laszlo Rikker, Xueming Yu, Ryan D. Burgert, Ning Yu, and Paul E. Debevec. Diffrelight: Diffusion-based facial performance relighting. In Takeo Igarashi, Ariel Shamir, and Hao (Richard) Zhang, editors, *SIGGRAPH Asia 2024 Conference Papers, SA 2024, Tokyo, Japan, December 3-6, 2024*, pages 11:1–11:12. ACM, 2024.

[46] Xianglong He, Junyi Chen, Sida Peng, Di Huang, Yangguang Li, Xiaoshui Huang, Chun Yuan, Wanli Ouyang, and Tong He. Gvgen: Text-to-3d generation with volumetric representation. In *European Conference on Computer Vision*, pages 463–479. Springer, 2024.

[47] Yuze He, Yushi Bai, Matthieu Lin, Wang Zhao, Yubin Hu, Jenny Sheng, Ran Yi, Juanzi Li, and Yong-Jin Liu. T3bench: Benchmarking current progress in text-to-3d generation. *arXiv preprint arXiv:2310.02977*, 2023.

[48] Zhenliang He, Meina Kan, and Shiguang Shan. Eigengan: Layer-wise eigen-learning for gans. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14408–14417, 2021.

[49] Zhuo He, Paul Henderson, and Nicolas Pugeault. Beyond reconstruction: A physics based neural deferred shader for photo-realistic rendering. *arXiv preprint arXiv:2504.12273*, 2025.

[50] Martin Heusel et al. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems*, page 6629–6640, 2017.

[51] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[52] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, pages 6840–6851, 2020.

[53] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 6840–6851, 2020.

[54] Fangzhou Hong, Jiaxiang Tang, Ziang Cao, Min Shi, Tong Wu, Zhaoxi Chen, Shuai Yang, Tengfei Wang, Liang Pan, Dahua Lin, et al. 3dtopia: Large text-to-3d generation model with hybrid diffusion priors. *arXiv preprint arXiv:2403.02234*, 2024.

[55] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.

[56] Xin Huang, Tengfei Wang, Ziwei Liu, and Qing Wang. Material anything: Generating materials for any 3d object via diffusion. *CoRR*, abs/2411.15138, 2024.

[57] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[58] Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. Neural wavelet-domain diffusion for 3d shape generation. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022.

[59] Arthur Jacot et al. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018.

[60] Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.

[61] Nikita Jaipuria, Shubh Gupta, Praveen Narayanan, and Vidya N Murali. On the role of receptive field in unsupervised sim-to-real image translation. *arXiv preprint arXiv:2001.09257*, 2020.

[62] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5322–5332, 2024.

[63] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

[64] James T. Kajiya. The rendering equation. In *SIGGRAPH*, pages 143–150, 1986.

[65] James T. Kajiya. The rendering equation. *ACM SIGGRAPH Computer Graphics*, 20(4):143–150, 1986.

[66] Brian Karis and Epic Games. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 4(3):1, 2013.

[67] Animesh Karnewar, Niloy J Mitra, Andrea Vedaldi, and David Novotny. Holofusion: Towards photo-realistic 3d generative modeling. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22976–22985, 2023.

[68] Animesh Karnewar, Andrea Vedaldi, David Novotny, and Niloy J Mitra. Holodiffusion: Training a 3d diffusion model using 2d images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18423–18433, 2023.

[69] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020.

[70] Tero Karras et al. A Style-Based Generator Architecture for Generative Adversarial Networks. *Conference on Computer Vision and Pattern Recognition*, 2019.

[71] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[72] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[73] Bernhard Kerbl et al. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, pages 1–14, August 2023.

[74] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[75] Johannes Kerbl, Mathias Wacker, Lars Hansen, Rasmus Jensen, and Vincent Vanhoucke. Fast gaussian splatting for real-time radiance field rendering. In *SIGGRAPH*, 2023.

[76] Hoel Kervadec, Jose Dolz, Chao Wang, Eric Granger, Ismail Ben Ayed, and Christian Desrosiers. Boundary loss for highly unbalanced segmentation. *arXiv preprint arXiv:1812.07032*, 2019.

[77] Hoon Kim et al. SwitchLight: Co-Design of Physics-Driven Architecture and Pre-training Framework for Human Portrait Relighting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25096–25106. IEEE, June 2024.

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

[79] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

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

[81] Tobias Kirschstein, Simon Giebenhain, and Matthias Nießner. Diffusionavatars: Deferred diffusion for high-fidelity 3d head avatars. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 5481–5492. IEEE, 2024.

[82] Black Forest Labs et al. FLUX.1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space. *arXiv preprint arXiv:2506.15742*, 2025.

[83] Yushi Lan, Fangzhou Hong, Shuai Yang, Shangchen Zhou, Xuyi Meng, Bo Dai, Xingang Pan, and Chen Change Loy. Ln3diff: Scalable latent neural fields diffusion for speedy 3d generation. In *European Conference on Computer Vision*, pages 112–130. Springer, 2024.

[84] Hung Le and Ali Borji. What are the receptive, effective receptive, and projective fields of neurons in convolutional neural networks? *arXiv preprint arXiv:1705.07049*, 2017.

[85] Thomas Leimkuhler, Kalyan Sunkavalli, and Pradeep Sen. Neural deferred shading for real-time rendering. In *SIGGRAPH Asia*, 2020.

[86] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *arXiv preprint arXiv:2311.06214*, 2023.

[87] Tzu-Mao Li, Miika Aittala, Frédo Durand, Ravi Ramamoorthi, and Wojciech Matusik. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics*, 39(4):124:1–124:17, 2020.

[88] Tzu-Mao Li, Miika Aittala Li, Frédo Durand, Wojciech Matusik, and Ravi Ramamoorthi. Differentiable monte carlo ray tracing through edge sampling. In *TOG (SIGGRAPH)*, volume 37, pages 222:1–222:13, 2018.

[89] Weiyu Li, Jiarui Liu, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. *arXiv preprint arXiv:2405.14979*, 2024.

[90] Yuheng Li, Haotian Liu, Qingyan Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521, 2023.

[91] Zejian Li, Yongchuan Tang, Wei Li, and Yongxing He. Learning disentangled representation with pairwise independence. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4245–4252, 2019.

[92] Zhengqi Li and Noah Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In *Proceedings of the European conference on computer vision (ECCV)*, pages 371–387, 2018.

[93] Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Zhi-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, and Zian Wang. Diffusionrenderer: Neural inverse and forward rendering with video diffusion models. *arXiv preprint arXiv:2501.18590*, 2025.

[94] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6517–6526, 2024.

[95] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21644–21653, 2024.

[96] Chenguo Lin, Panwang Pan, Bangbang Yang, Zeming Li, and Yadong Mu. Diffsplat: Repurposing image diffusion models for scalable gaussian splat generation. *arXiv preprint arXiv:2501.16764*, 2025.

[97] Weifeng Lin, Xinyu Wei, Ruichuan An, Peng Gao, Bocheng Zou, Yulin Luo, Siyuan Huang, Shanghang Zhang, and Hongsheng Li. Draw-and-understand: Leveraging visual prompts to enable mllms to comprehend what you want. *CoRR*, abs/2403.20271, 2024.

[98] Yehonathan Litman, Or Patashnik, Kangle Deng, Aviral Agrawal, Rushikesh Zawar, Fernando De la Torre, and Shubham Tulsiani. Materialfusion: Enhancing inverse rendering with material diffusion priors. *arXiv preprint arXiv:2409.15273*, 2024.

[99] Yehonathan Litman, Or Patashnik, Kangle Deng, Aviral Agrawal, Rushikesh Zawar, Fernando De la Torre, and Shubham Tulsiani. Materialfusion: Enhancing inverse rendering with material diffusion priors. In *3DV*, 2025.

[100] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10072–10083, 2024.

[101] Shichen Liu, Tianye Li, Weikai Chen, Jingwan Liao, Ming-Hsuan Yang, Xiaowei Lu, and Hao Hu. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *ICCV*, pages 7708–7717, 2019.

[102] Matthew Loper and Michael J. Black. Opendr: An approximate differentiable renderer. In *ECCV*, pages 154–169, 2014.

[103] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2837–2845, 2021.

[104] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *Advances in Neural Information Processing Systems*, 36:75307–75337, 2023.

[105] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

[106] Yupeng Ma, Xiaoyi Feng, Xiaoyue Jiang, Zhaoqiang Xia, and Jinye Peng. Intrinsic image decomposition: A comprehensive review. In *Image and Graphics: 9th International Conference, ICIG 2017, Shanghai, China, September 13-15, 2017, Revised Selected Papers, Part I 9*, pages 626–638. Springer, 2017.

[107] David McAllister, Songwei Ge, Jia-Bin Huang, David Jacobs, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Rethinking score distillation as a bridge between image distributions. *Advances in Neural Information Processing Systems*, 37:33779–33804, 2024.

[108] Ben Mildenhall et al. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, pages 405–421, Berlin, Heidelberg, August 2020.

[109] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–421, 2020.

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

[111] Chong Mou, Xintao Wang, Jie Song, Jian Chen, Jian Bai, Songhai He, and Ying Shan. T2i-adapter: Learning adapters to inject controls into text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6816–6826, 2023.

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

[113] Martin E. Newell and James F. Blinn. The progression of realism in computer generated images. In *Proceedings of the 1977 annual conference on - ACM '77*, pages 444–448, 1977.

[114] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

[115] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7184–7193, 2019.

[116] Yotam Nitzan, Amit Bermano, Yangyan Li, and Daniel Cohen-Or. Face identity disentanglement via latent space mapping. *arXiv preprint arXiv:2005.07728*, 2020.

[117] Hao Ouyang, Kathryn Heal, Stephen Lombardi, and Tiancheng Sun. Text2immersion: Generative immersive scene with 3d gaussians. *arXiv preprint arXiv:2312.09242*, 2023.

[118] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to-image synthesis. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[119] Matt Pharr and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2016.

[120] Yohan Poirier-Ginter, Alban Gauthier, Julien Phillip, J-F Lalonde, and George Drettakis. A diffusion approach to radiance field relighting using multi-illumination

synthesis. In *Computer Graphics Forum*, volume 43, page e15147. Wiley Online Library, 2024.

[121] An-Chieh Ponce, Yu-Hsuan Chang, Wei-Chen Chiu, and Hung-Yu Liu. Dilightnet: Fine-grained dibr-based illumination for text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7406–7415, 2023.

[122] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[123] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations*, 2016.

[124] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. *SIGGRAPH*, pages 117–128, 2001.

[125] Aditya Ramesh, Youngduck Choi, and Yann LeCun. A spectral regularizer for unsupervised disentanglement. *CoRR*, abs/1812.01161, 2018.

[126] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

[127] Stephan R. Richter et al. Enhancing Photorealism Enhancement. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1700–1715, February 2023.

[128] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2022.

[129] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[130] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE, 2022.

[131] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[132] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha G Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[133] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.

[134] Subhransu M. Sengupta, Federico M. Carlucci, Xiaohui Ru, Le Duan, Vedika Madhavan, and Wojciech Matusik. Neural inverse rendering of an indoor scene from a single image. In *SIGGRAPH*, 2021.

[135] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021.

[136] Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Transactions on Graphics (TOG)*, 42(4):1–16, 2023.

[137] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023.

[138] Yahao Shi, Yanmin Wu, Chenming Wu, Xing Liu, Chen Zhao, Haocheng Feng, Jian Zhang, Bin Zhou, Errui Ding, and Jingdong Wang. Gir: 3d gaussian inverse rendering for relightable scene factorization. *arXiv preprint arXiv:2312.05133*, 2023.

[139] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

[140] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3d generation. In *The Twelfth International Conference on Learning Representations*, 2024.

[141] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20875–20886, 2023.

[142] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[143] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

[144] Stanislaw Szymanowicz et al. Splatter Image: Ultra-Fast Single-View 3D Reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10208–10217, 2024.

[145] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Viewset diffusion:(0-) image-conditioned 3d generative models from 2d data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8863–8873, 2023.

[146] Matthew Tancik et al. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *Advances in Neural Information Processing Systems*, pages 7537–7547, 2020.

[147] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024.

[148] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.

[149] Zhicong Tang, Shuyang Gu, Chunyu Wang, Ting Zhang, Jianmin Bao, Dong Chen, and Baining Guo. Volumediffusion: Flexible text-to-3d generation with efficient volumetric encoder. *arXiv preprint arXiv:2312.11459*, 2023.

[150] A. Tewari et al. Advances in Neural Rendering. *Computer Graphics Forum*, pages 703–735, 2022.

[151] Justus Thies et al. Deferred neural rendering: image synthesis using neural textures. *ACM Trans. Graph.*, July 2019.

[152] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024.

[153] Eric Veach and Leonidas J Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428, 1995.

[154] Bruce Walter et al. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, EGSR'07, June 2007.

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

[156] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4563–4573, 2023.

[157] Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, and Jifeng Dai. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *CoRR*, abs/2411.10442, 2024.

[158] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. In *European conference on computer vision*, pages 57–74. Springer, 2024.

[159] Markus Worchel et al. Multi-View Mesh Reconstruction with Neural Deferred Shading. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6177–6187, June 2022.

[160] Tong Wu, Jia-Mu Sun, Yu-Kun Lai, Yuewen Ma, Leif Kobbelt, and Lin Gao. Deferredgs: Decoupled and editable gaussian splatting with deferred shading. *arXiv preprint arXiv:2404.09412*, 2024.

[161] Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, Zhenda Xie, Yu Wu, Kai Hu, Jiawei Wang, Yaofeng Sun, Yukun Li, Yishi Piao, Kang Guan, Aixin Liu, Xin Xie, Yuxiang You, Kai Dong, Xingkai Yu, Haowei Zhang, Liang Zhao, Yisong Wang, and

Chong Ruan. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *CoRR*, abs/2412.10302, 2024.

[162] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12863–12872, 2021.

[163] Dejia Xu, Ye Yuan, Morteza Mardani, Sifei Liu, Jiaming Song, Zhangyang Wang, and Arash Vahdat. Agg: Amortized generative 3d gaussians for single image to 3d. *arXiv preprint arXiv:2401.04099*, 2024.

[164] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.

[165] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

[166] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. In *European Conference on Computer Vision*, pages 1–20. Springer, 2024.

[167] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *arXiv preprint arXiv:2311.09217*, 2023.

[168] Bowen Xue, Claudio Guarnera, Shuang Zhao, and Zahra Montazeri. Reflectancefusion: Diffusion-based text to svbrdf generation. In *Eurographics Symposium on Rendering*. Eurographics Association, 2024.

[169] Yao Yao et al. Neilf: Neural incident light field for physically-based material estimation. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, page 700–716, 2022.

[170] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021.

[171] Keyang Ye et al. 3D Gaussian Splatting with Deferred Reflection. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24*, pages 1–10, July 2024.

[172] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6796–6807, 2024.

[173] Guan Yin, Zhaoyang Yuan, Yasasa Li, Ling-Hao Liu, Yi Zhou, Tiezheng Wang, He Wang, and Guanying Wang. Diffusionlight: Light-aware stylization with universal relighting. *arXiv preprint arXiv:2312.06148*, 2023.

[174] Alex Yu et al. pixelNeRF: Neural Radiance Fields From One or Few Images. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587, 2021.

[175] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations (ICLR)*, 2016.

[176] Zehao Yu et al. Mip-Splatting: Alias-free 3D Gaussian Splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19447–19456, 2024.

[177] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

[178] Tizian Zeltner* et al. Real-time Neural Appearance Models. *ACM Trans. Graph.*, pages 33:1–33:17, June 2024.

[179] Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. Rgb↔x: Image decomposition and synthesis using material- and lighting-aware diffusion models. `https://github.com/zheng95z/rgbx`, 2024. ACM SIGGRAPH 2024; accessed 2026-05-04.

[180] Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. Gaussiancube: Structuring gaussian splatting using optimal transport for 3d generative modeling. *arXiv e-prints*, pages arXiv–2403, 2024.

[181] Hantao Zhang. Towards principled methods for training generative adversarial networks.

[182] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 43(4):1–20, 2024.

[183] Lvmin Zhang and Anyi Mano. Scaling in-the-wild training for diffusion-based illumination harmonization and editing by imposing consistent light transport. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[184] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 3813–3824. IEEE, 2023.

[185] Richard Zhang et al. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, Salt Lake City, UT, June 2018.

[186] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[187] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. Faceboxes: A cpu real-time face detector with high accuracy. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9. IEEE, 2017.

[188] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. S3fd: Single shot scale-invariant face detector. In *Proceedings of the IEEE international conference on computer vision*, pages 192–201, 2017.

[189] Yuqing Zhang, Yuan Liu, Zhiyu Xie, Lei Yang, Zhongyuan Liu, Mengzhou Yang, Runze Zhang, Qilong Kou, Cheng Lin, Wenping Wang, et al. Dreammat: High-quality pbr material generation with geometry-and light-aware diffusion models. *ACM Transactions on Graphics (TOG)*, 43(4):1–18, 2024.

[190] Yuqing Zhang, Yuan Liu, Zhiyu Xie, Lei Yang, Zhongyuan Liu, Mengzhou Yang, Runze Zhang, Qilong Kou, Cheng Lin, Wenping Wang, and Xiaogang Jin. Dreammat: High-quality PBR material generation with geometry- and light-aware diffusion models. *ACM Trans. Graph.*, 43(4):39:1–39:18, 2024.

[191] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57, 2016.

[192] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z Li. Face alignment in full pose range: A 3d total solution. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):78–92, 2017.

[193] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa volume splatting. In *Proceedings Visualization, 2001. VIS'01.*, pages 29–538. IEEE, 2001.