Diffusion-based photo-realistic rendering with scalable synthetic-real paired data

Anonymous Author(s)
Affiliation
Address
email

Abstract

Photorealistic rendering aims to produce images indistinguishable from real-world photographs. Traditional rendering techniques, while effective, rely heavily on synthetic models with intricate material properties. While neural rendering methods offer a potential solution to this problem, they often necessitate data from costly capturing equipment like the Light Stage or resort to low-quality synthetic data, hindering their ability to achieve photo-realistic rendering. To address these challenges, we propose a novel image rendering framework including a data generation method and a neural rendering model. Our data generation method can create synthetic-real data pairs using intrinsic decomposition methods, leveraging intrinsic images similar to G-buffers in the traditional rendering pipeline. Additionally, we introduce a photorealistic image synthesis method based on diffusion models, enhancing the generalization capabilities of our framework. This framework allows for scalable data generation and photorealistic rendering for low-quality synthetic objects. Experiments show that our method can not only render comparable images with sophisticated synthetic 3D models but can fulfill state-of-the-art rendering for low-quality synthetic 3D models.

1 Introduction

In the pursuit of photorealistic rendering, the goal is to generate images that are virtually indistinguishable from real-world photographs, a necessity in fields like game production and immersive virtual reality. With the evolution of computer hardware, the advent of physically based rendering (PBR) utilizing recursive path tracing has become feasible. Over recent decades, researchers have dedicated significant efforts to crafting sophisticated rendering models[41][53][5] that meticulously account for lighting, materials, and object geometry to achieve photorealism. These rendering techniques are now commonplace in production engines such as Unreal Engine[15] and Unity[52]. However, their efficacy often hinges on the availability of high-quality CAD models with intricate material properties, necessitating extensive manual labor. Consequently, rendering photorealistic images remains a challenge when dealing with low-quality CAD models generated through manual design or techniques like Multi-View-Stereo (MVS).

With the advent of deep learning, the prospect of generating photorealistic images using end-to-end neural networks has become a reality. Previous research[51][55][40] has utilized reflectance fields obtained from Light Stage[11][12] captures to construct datasets, which are then employed to train rendering networks. Leveraging these high-quality datasets, neural networks have achieved state-of-the-art results. However, acquiring such datasets poses a challenge for consumers, as it typically requires access to expensive Light Stage equipment. To circumvent this limitation, some methodologies[32][70][61] have emerged to synthesize training datasets. While these approaches prove effective to a certain extent, networks trained solely on synthetic datasets often struggle to

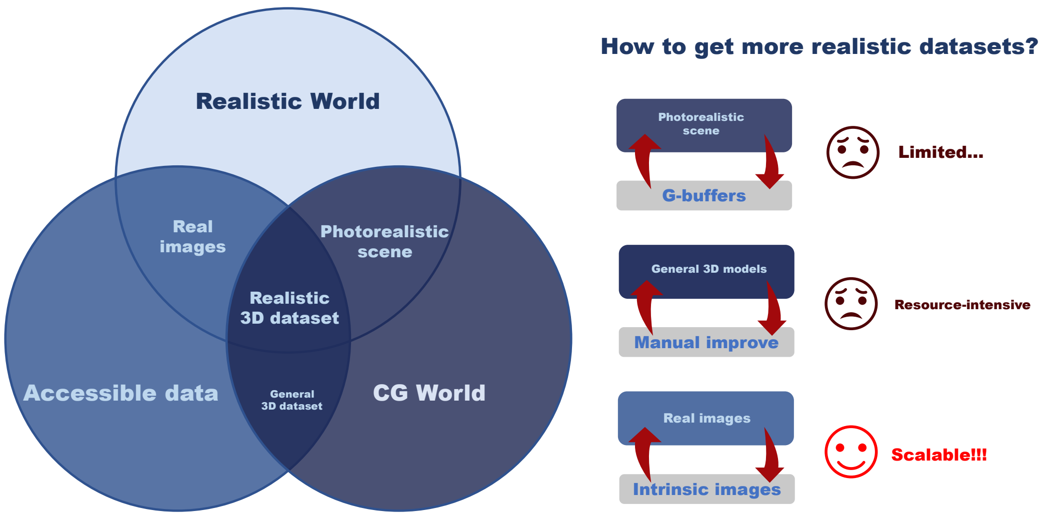

Figure 1: The teaser shows our major contribution to solving the problem of lacking realistic paired images for neural rendering. According to the figure on the left side, we can enrich the realistic paired images from three different resources - Photorealistic 3D scenes, general 3D datasets, and real images. Though there is a large amount of photorealistic 3D scenes in film productions that can be used to render high-quality paired datasets, it is difficult to get access to them. While manually creating more realistic data may seem intuitive, it is a resource-intensive process, demanding significant time and financial investment. Numerous general 3D datasets exist, but their quality may not be as satisfactory. Compared to these two resources, generating realistic paired data from real images is promising since the novel framework we proposed, based on intrinsic images and the neural rendering method, has proved to be practical.

generalize well to real-world scenarios, hindering their ability to render photorealistic images for synthetic objects.

In response to the challenge of limited paired data for rendering synthetic objects with photorealism, we present a novel data generation pipeline capable of producing synthetic-real data pairs. Our key insight stems from the observation that intrinsic images derived from intrinsic decomposition methods exhibit a domain similarity to the G-buffers within the computer graphics (CG) rendering pipeline, as shown in Figure 2. Specifically, we extract irradiance images, specular shading images, and albedo images as the source data, as they encompass low-frequency lighting information, high-frequency view-dependent information, and identity information, respectively. To obtain these images from natural scenes, we employ the IntrinsicAnything[9] method which is trained on the synthetic Objaverse dataset[13]. While IntrinsicAnything excels in generating high-quality albedo images and specular shading images, it lacks the capability to produce irradiance images. To address this limitation, we introduce a novel method designed to generate all aforementioned intrinsic components.

With the generated data, we further propose a photorealistic image synthesis method based on the diffusion model. Recently, the diffusion model[19][46] has demonstrated state-of-the-art performance across various tasks[24][36][33][16]. These methodologies have shown that diffusion models, even when fine-tuned on small datasets, exhibit strong generalization capabilities. This is primarily attributed to the robust priors learned by diffusion models from extensive real data. In this paper, we also capitalize on the strength of the diffusion prior. Specifically, we first use pre-trained visual encoders to extract global and local features of the intrinsic images. Then these features are injected into a pre-trained text-to-image diffusion model as guidance.

Through our data generation pipeline, we can readily scale up the training dataset by gathering extensive natural images from the internet and processing them. This scalability is pivotal for enhancing the generalization capabilities of neural networks. Moreover, leveraging the robust diffusion prior, our fine-tuned diffusion model excels not only in rendering comparable images for high-quality CAD models but also in generating photorealistic images for even the most low-quality synthetic objects. In summary, our main contributions are:

Figure 2: Visualization of the G-buffers and intrinsic images.

- We propose a novel training framework for photo-realistic image synthesis which includes a data generation method and a neural rendering model.
- We achieve state-of-the-art photorealistic rendering results.

2 Related work

2.1 Photorealistic Rendering

Modern graphics heavily rely on Physically Based Rendering (PBR) techniques to achieve photorealism. PBR methods[41][4][53][5], incorporating lighting, materials, and geometry, render meticulously crafted CAD models according to the rendering equation. Early graphic production saw the prominence of models like the Phong[41] and its variant, the Blinn-Phong[4] model, which delivered a commendable performance. However, sophisticated material properties were necessitated for more realistic rendering, leading to the development of advanced models like the Cook-Torance GGX[53] and Disney GGX[5]. While these PBR methods enable photo-realistic rendering, they demand high-quality CAD models with intricate materials and geometry, incurring substantial costs. To address this challenge, Paul Debevec *et al.* introduced the Light Stage[12], enabling the capture of reflectance fields for human portraits, facilitating rendering under any natural environment[10]. Although combining reflectance fields with environment maps can produce nearly natural images, this approach is limited to subjects with pre-captured reflectance fields. Moreover, accessibility to such technology remains restricted, constraining its widespread application.

Another significant avenue of photorealistic rendering is neural rendering, which harnesses the power of neural networks to generate high-quality images. A common task in neural rendering is relighting, wherein the goal is to alter the appearance of subjects in an image to match a different target environment light. Existing learning-based approaches either[51][55][40][43][67][25] utilize images generated from reflectance fields[11] or synthesized datasets[32][70][61] to train end-to-end networks for single image relighting. Some methods[51][70][43][67] alter the light information and re-render input images fully in the latent space, while others[32][55][40][61] first estimate intrinsic image properties before combining them with target lighting for neural rendering. SIPR[51] and DSIPR[70] inject target lighting into the network bottleneck and use the decoder for rendering. PhotoApp[43] edits lighting in the latent space of StyleGAN[23], then decodes the latent code for rendering. NVPR[67] introduces self-supervised losses to disentangle lighting and identity features, which helps to render with novel light. Li et al.[32] design a multi-bounce scheme to ease the problem of not considering the global illumination and a cascade framework for narrowing the errors of the predictions. Thanks to the existence of the shadow maps and specular maps in the generated dataset, Wang et al. [55] design the SS network to predict the specular maps and shadows maps which are input into the composition network together with light and albedo to composite a realistic rendering image. Total Relighting[40] utilizes convolved environment maps and normal maps to generate light maps containing explicit lighting cues. These light maps, along with other intrinsic properties, serve as inputs to a shading network for rendering photorealistic images. Due to the absence of publicly available light stage datasets, Lumos[61] relies on purchased 3D face scans to generate large-scale relighting pairs. Following the Total Relighting framework, Lumos initially trains networks using synthetic datasets and then refines the results by learning a residual map that minimizes the domain gap between synthetic and real albedo using a real dataset. By leveraging extracted image intrinsics, SwitchLight[25] combines the Cook-Torrance model for initial relighting with a neural network for enhanced refinement. Recently, diffusion models[19][46] trained on large-scale natural images have demonstrated impressive performance across various vision tasks. To fully exploit the priors embedded in diffusion models for rendering, several recent works[14][42][27][63] have made notable attempts. DiffusionRig[14] and DiFaReli[42] first utilize DECA to predict FLAME[30] parameters and spherical harmonics (SH) light information. Subsequently, they render physical buffers containing light information to guide the diffusion process. LightIt[27] predicts shading maps for outdoor scenes, utilizing them as conditions for generating relighted images. DiLightNet[63], by pre-defining multiple roughness levels, generates radiance hints as guidance for the diffusion model. In our paper, we aim to condition the diffusion model on albedo, irradiance, and specular shading maps to achieve photo-realistic rendering.

2.2 Inverse Rendering

Inverse rendering endeavors to recover the intrinsic properties (such as geometry, materials, and lighting) of a scene or object, either in 2D[17][28][20][31][62][59][35][58][56] or 3D[7][57][1][54][66][50][68][65][69][29][21][64][34][38] space. In 3D inverse rendering, some methods[7][1][54][57] address the problem based on explicit mesh. DIB-R++[7] first predicts the mesh with UV mapping from the input single image, then optimizes the material textures and lighting via the differentiable renderer. Azinovic *et al.*[1] and SunStage[54] estimate the material textures based on the reconstructed FLAME mesh. FIPT[57] calculates the pre-baked shading maps with the provided or optimized mesh and uses the shading maps for materials estimation. Others[50][68][29][66][65][69][34][38] optimize all the intrinsics in implicit neural radiance field. To adapt Nerf[37] for inverse rendering, NERV[50], NerFactor[68] and NEROIC[29] assign materials for each sample point and calculate the outgoing radiance of them with physically-based rendering (PBR). These models[66][65][69][34][38] represent geometry as signed distance function (SDF) and optimize the neural materials of the implicit surfaces.

While inverse rendering in 3D has achieved promising results, it often suffers from computational inefficiencies due to the utilization of computation-intensive Multi-Layer Perceptron (MLP). To mitigate this issue, single-image inverse rendering focuses on recovering intrinsic properties using more generalizable neural networks. David *et al.* [17] treats the albedo estimation task as a light diffusion process and iteratively diffuses the image to get albedo. IID[28], trained on indoor synthetic datasets, utilizes diffusion priors to estimate material maps. Due to the lack of ground truth material maps, unsupervised methods[20][31][62][59][35][58][56] estimate the intrinsics based on hand-crafted priors.

3 Method

Given G-buffers or intrinsic images of an object, our task is to render photo-realistic images. Figure 1 shows the overview of our framework. Our innovation is in a novel rendering framework that includes a novel synthetic-real paired data generation method (Section 3.1) and a rendering method leveraging strong diffusion priors. To render photo-realistic images with G-buffers or intrinsic images, our core idea is to extract the global identity features (Section 3.2.1) and the local detailed features (Section 3.2.2), then inject (Section 3.3) them into a pre-trained text-to-image diffusion model.

3.1 Data generation

Though neural rendering shows great performance in fulfilling photo-realism, the data is hard to collect. To solve this problem, we are the first to generate intrinsic-real data pairs for training. Our key observation is that the intrinsic images estimated by intrinsic decomposition methods share a similar domain to the G-buffers of synthetic CAD models. Thus, we can treat the intrinsic domain as a proxy domain, the images from which can be rendered or mapped to natural images. After training, we can obtain photo-realistic images from the G-buffers of the synthetic CAD models.

To get the intrinsic images of natural images, we take the state-of-the-art intrinsic decomposition method IntrinsicAnything[9] as the baseline, which can generate the albedo image and specular shading image from a natural image. However, these two images cannot fully represent the information present in an image, as they only contain the identity and high-frequency lighting information

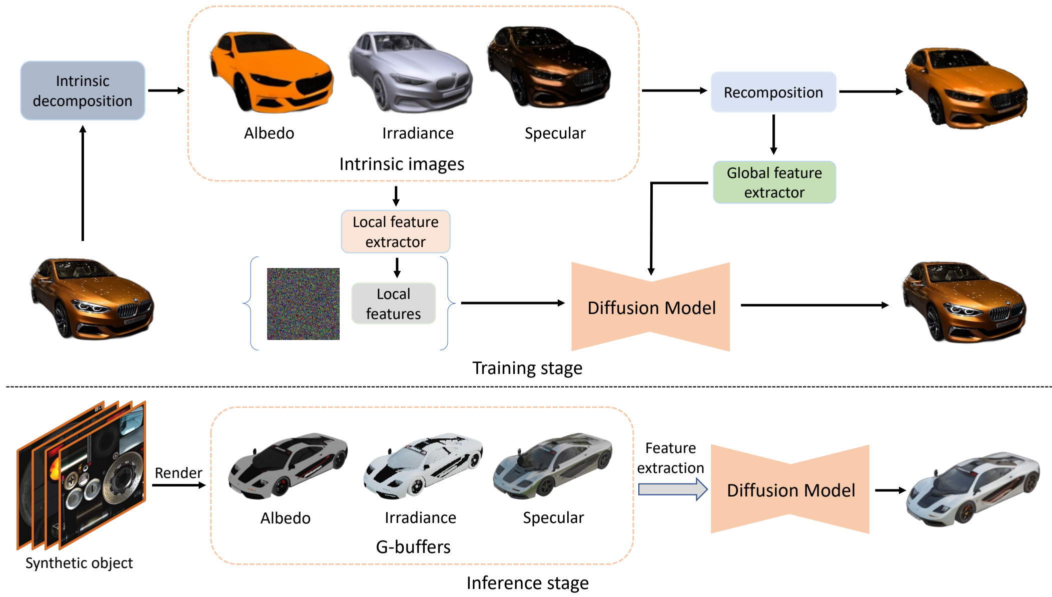

Figure 3: **Overview of our framework.** Given natural images, we first generate intrinsic-real data pairs for training. At the training stage, we extract the global features by encoding the recomposed image into a pre-trained self-supervised visual encoder and extract the local features with a pre-trained VAE encoder. We then inject the features into a pre-trained text-to-image diffusion model for rendering. At the inference stage, we extract features of the rendered G-buffers of the synthetic CAD model, and then inject them into the model for photo-realistic rendering.

respectively. To represent the low-frequency lighting information, we need another component, i.e., the irradiance image. So the forward pass of intrinsic image generation is defined as:

$$A, S, I = F_\theta(I) \tag{1}$$

where A, S and I are the albedo image, specular shading image, and irradiance image respectively, F is the intrinsic decomposition model and θ is the model parameter.

To adapt IntrinsicAnything for irradiance estimation, we first render irradiance images for Objaverse dataset, then we train another diffusion model using these images for irradiance estimation.

3.2 Feature extraction

Previous visual encoders such as CLIP[44] and Dino[6] are built on natural images. Directly applying them on intrinsic images may fail to extract representative global features. To address this problem, we propose to recompose the intrinsic images into a natural image according to the rendering equation. Then we use the self-supervised model DINOv2[39] to extract the global identity features. As for the local detailed features, we use the pre-trained VAE encoder to extract the pixel-aligned detail features.

3.2.1 Global feature extraction

Image recomposition. The rendering equation can be represented as:

$$L = L_d + L_s \tag{2}$$

where L_d is the diffuse shading and L_s is the specular shading. The diffuse shading can further be represented as:

$$L_d = A * I \tag{3}$$

where A is the albedo image and I is the irradiance image. So we can get the recomposed image as:

$$R(x) = A(x) * I(x) + S(x) \tag{4}$$

where x is the pixel coordinate of the image, $A(x)$, $I(x)$ and $S(x)$ are the albedo image, irradiance image, and specular shading image at x respectively. Firstly, we transform these images into linear space. Then we compose them according to Equation 4. Finally, we transform the linear-space image back to sRGB space by gamma correction. Though the recomposed image loses some high-dynamic-range information compared to the source natural image due to truncation, we find that the extracted features are robust enough for rendering.

Feature extraction. Previous works[49][60] use the CLIP image encoder to extract the global features. However, the CLIP image embedding is aligned with the text embedding. Text, being a coarse description, is often insufficient to represent intricate details. Inspired by Anydoor[8], we choose the Dinov2 as our global feature extractor which encodes the image as a global token and a patch token. Following AnyDoor, we concatenate the two tokens and use a linear layer to project the tokens to the diffusion embedding space. The process is defined as:

$$t_l, t_g = F_G(R) \tag{5}$$

$$f_g = L(Con(t_l, t_g)) \tag{6}$$

where R is the recomposed image, F_G is the Dinov2 model, t_l is the patch token, t_g is the global token, Con means concatenation along channel dimension, L is the linear projection layer and f_g is the global feature in the diffusion embedding space.

3.2.2 Local feature extraction

Global features encode the identity of the object, while local features encode the details of the object. Though the VAE model of Stable Diffusion is also trained on natural images, these methods[24][16][36] proved that they can extract features rich in detailed information for intrinsic images such as the normal map and the depth map. Thus, we use the pre-trained VAE encoder to extract the pixel-aligned detail features. Specifically, we first use the VAE encoder to encode the three intrinsic images. Then we concatenate the three feature maps along the channel dimension. The concatenated feature map will further be concatenated with the noise image as the input to the Unet encoder.

The process is defined as:

$$f_a, f_i, f_s = F_L(A, I, S) \tag{7}$$

$$f_l = Con(f_a, f_i, f_s) \tag{8}$$

where A, I, and S are the albedo image, irradiance image, and specular shading image respectively, F_L is the pre-trained VAE encoder, f_a, f_i and f_s are the albedo feature map, irradiance feature map and specular shading feature map respectively, and Con means concatenation along channel dimension.

3.3 Feature injection

After obtaining the global and local features, we inject them into a pre-trained text-to-image diffusion model. In our paper, we take the Stable Diffusion[47] as the backbone which has been demonstrated robust. We denote the diffusion model as ϵ_θ, so the training objective can be defined as:

$$\mathbb{E}_{t,\mathbf{x}_0,\epsilon,c}\left[\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t, c\right)\right\|_2\right] \tag{9}$$

where $\bar{\alpha}_t$ is the schedule parameter, $\mathbf{x}_0$ is the target image, ϵ is the noise image sampled from $N(0, I)$, t is the time step, and c is the extracted features. The training objective is to minimize the L2 distance between the predicted noise vector and the ground-truth sampled noise vector.

Specifically, we inject the global features into the diffusion model by replacing the text embedding with them and performing cross-attention between them and Unet intermediate features. As for the local detailed features, we concatenate them with the noise image as the input to the Unet encoder.

4 Experiments

4.1 Implement Details

We implement all our methods in Pytorch. We use the Adam[26] optimizer with an initial learning rate of $1e^{-5}$ and linearly decay the learning rate to 0. During each iteration, we take a batch size of 128 for training. We optimize our model for 20k iterations which takes 4 days on 8 Nvidia A800 (80GB) GPUs. Other hyperparameters for our network follow the default settings in the Stable Diffusion model.

4.2 Dataset and Metrics

Table 1: Comparison of FID, KID, and Inception Score on face and car datasets

Method	Face Dataset			Car Dataset		
	FID $\downarrow$	KID$\times10^3$ $\downarrow$	Inception Score $\uparrow$	FID $\downarrow$	KID$\times10^3$ $\downarrow$	Inception Score $\uparrow$
PBR	42.5	37.1 ± 1.2	3.88 ± 0.19	32.1	9.64 ± 0.09	$\mathbf{2.25} \pm \mathbf{0.19}$
Ours	$\mathbf{31.1}$	$\mathbf{25.0} \pm 1.2$	$\mathbf{4.18} \pm 0.18$	$\mathbf{28.9}$	$\mathbf{7.17} \pm \mathbf{0.06}$	2.23 ± 0.17

Datasets As mentioned, our training framework can generate synthetic-real paired data from real data. To evaluate the performance of the framework and the diffusion-based synthesis method, we conducted experiments on two real datasets: the real car dataset and the real face dataset. We collected the car dataset from the Internet. This dataset contains about 1000 cars, each of which has about 6 multi-view images. For the face dataset, we used the FFHQ dataset [22] that provides 70,000 high-quality face images in total. We split the car and face datasets so that ninety percent of them are used for training and the rest are used for evaluation. To prove the capability of generalization, we test our model on limited-quality synthetic face and car models collected from TURBOSQUID and CGTrader website. The links of these models are presented in the supplement material. Furthermore, we also evaluate our model on a high-quality synthetic dataset–hyperSim [45] that contains over 77400 photorealistic images of 461 indoor scenes with sophisticated and accurate lighting, materials, and geometry.

Metrics Metrics for evaluating the realism of the generated images have been widely used in the field of generative models. The most common metrics are Inception Score (IS) [48], Fréchet Inception Distance (FID) [18], and Kernel Inception Distance (KID) [2]. Inception score aims to measure the diversity and recognizability of generated images. It uses a pre-trained Inception network to classify generated images and calculates the score based on the entropy of the predicted class distribution. A high IS indicates that the generated images are both diverse and confidently classified into specific categories. However, IS does not consider the distribution difference between the generated images and the real images. FID assesses the similarity between the distributions of generated images and real images. It calculates the Fréchet distance between the feature vectors of real and generated images, extracted from a specific layer of a pre-trained Inception network. Lower FID values indicate that the generated images are closer in distribution to the real images. KID also evaluates the similarity between generated and real images but emphasizes semantic content. It uses a polynomial kernel to compute the Maximum Mean Discrepancy (MMD) between the feature representations of generated and real images, extracted from the third pooling layer (pool3) of a pre-trained Inception network. Lower KID values indicate better alignment between the distributions of generated and real images. It is widely accepted that these three metrics are complementary and should be used together to provide a comprehensive evaluation of the generated images.

Table 1 compares FID, KID, and Inception Score across different methods on the FFHQ dataset and our collected car dataset. Please note that PBR means directly computing the rendered results according to the recomposition method explained in Section 3.2.1 and KID has been shown with $\times1000$ for better readability. The gray part indicates the standard deviation of the metrics.

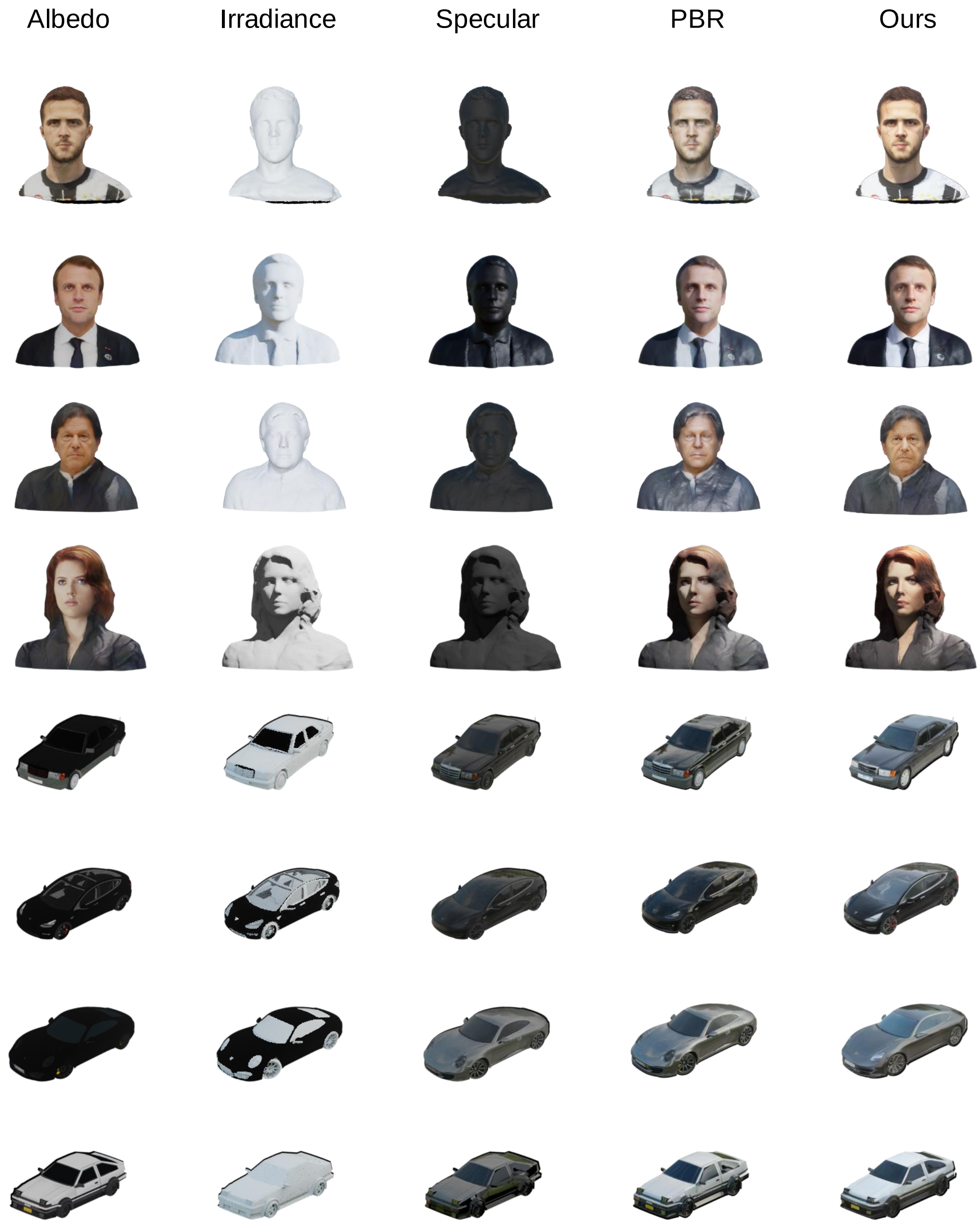

Figure 4: This figure shows some results of our diffusion-based method compared to the physically based rendering method on synthetic data. The models produced the face-related results trained on the FFHQ [22] dataset and the car-related results trained on our collected car dataset. Please note that face data have slight metallicity due to the capture process, which will result in the irregular specular effect present in the PBR-rendered images. However, our method can still generate realistic results.

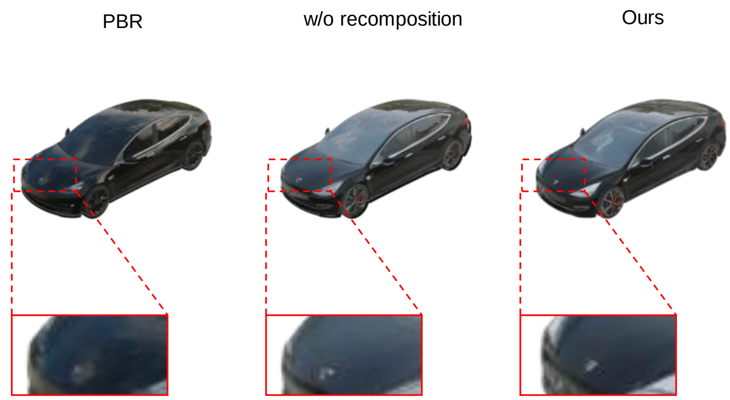

Figure 5: Ablation study for recomposition method.

4.3 Comparison with Baselines

Since our task aims to render photo-realistic images from G-buffers or intrinsic images, we compare our method with the physically based rendering (PBR) method. Specifically, we use the Principled BSDF node in Blender[3] as the description of the object's material and use environment maps as the light conditions for rendering. Our results show that we can achieve indistinguishable photo-realistic results compared to the baseline and even limited automatic realistic relighting. Some of our results are shown in Figure 4. Other test results on hyperSim[45] dataset are shown in the supplementary material, which are from .

4.4 Ablation Study

Recomposition The diffusion-based synthesis method is the core of our framework, and we conduct ablation studies to verify the effectiveness of the recomposition method. It is proved that the recomposition method can better guide the diffusion model. By recomposition, the generated images are more realistic and have better details. The examples are shown in the Figure 5. Due to the low-resolution of our output, the results may not be significantly different. However, the enlarged areas show kinds of details, which proves the effectiveness of our recomposition method.

Neural rendering method Before the hyper enthusiasm of diffusion-based image synthesis, there were also some neural rendering methods based on convolution networks. We also conduct this method on the above two datasets. The network used here is the Resnet Generator from the CycleGAN[71] method. The results in Figure 6 show that the results of face data are well satisfactory but those of cars are short of details and realism. Predictably, the network could perform well with enough data (the FFHQ dataset has nearly 70000 images, and the real car dataset only has 6000 images), which also means lots of data-driven methods could be progressed with our scalable data generation method.

5 Limitations

Our method relies highly on the quality of the generated data which is not guaranteed. In practice, we can find noisy data generated by the intrinsic decomposition method. Using these data for training may degrade the performance of the model. Meanwhile, the resolution of the generated intrinsic images is low due to the limitation of the intrinsic model, preventing the attainment of high-quality

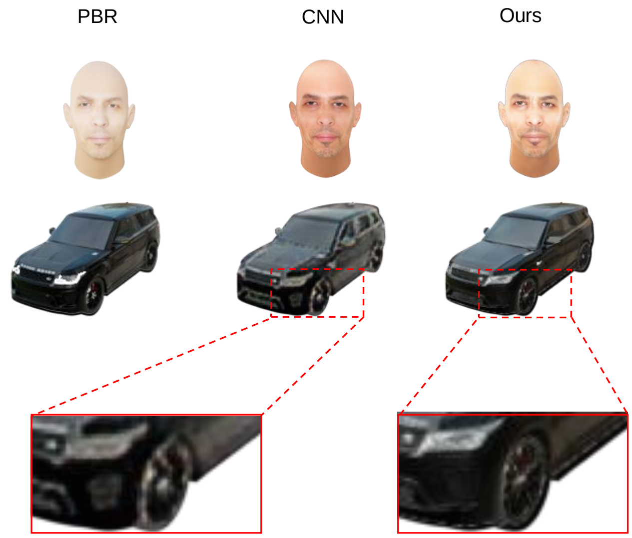

Figure 6: Ablation study for difference neural rendering methods.

results. This remains a future work for the intrinsic decomposition task. Besides, our method is based on the diffusion model, which is computationally expensive and not suitable for real-time rendering.

6 Conclusion

In this paper, we introduce a novel framework for photo-realistic rendering. This framework includes a novel synthetic-real paired data generation method and a diffusion-based neural rendering method. The data generation method leverages the intrinsic decomposition method to generate intrinsic images for real data. With these data, we extract their global and local features, and then inject them into a pre-trained text-to-image diffusion model. Qualitative and quantitative experiments demonstrate the effectiveness of our framework in generating photo-realistic images.

References

[1] Dejan Azinović, Olivier Maury, Christophe Hery, Matthias Nießner, and Justus Thies. High-res facial appearance capture from polarized smartphone images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16836–16846, 2023.

[2] Marcin Binkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[3] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2023.

[4] James F. Blinn. Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph.*, 11(2):192–198, jul 1977.

[5] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *Acm Siggraph*, volume 2012, pages 1–7. vol. 2012, 2012.

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[7] Wenzheng Chen, Joey Litalien, Jun Gao, Zian Wang, Clement Fuji Tsang, Sameh Khamis, Or Litany, and Sanja Fidler. Dib-r++: learning to predict lighting and material with a hybrid

differentiable renderer. *Advances in Neural Information Processing Systems*, 34:22834–22848, 2021.

[8] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. *arXiv preprint arXiv:2307.09481*, 2023.

[9] Xi Chen, Sida Peng, Dongchen Yang, Yuan Liu, Bowen Pan, Chengfei Lv, and Xiaowei Zhou. Intrinsicanything: Learning diffusion priors for inverse rendering under unknown illumination, 2024.

[10] Paul Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, page 189–198, New York, NY, USA, 1998. Association for Computing Machinery.

[11] Paul Debevec. The light stages and their applications to photoreal digital actors. *SIGGRAPH Asia*, 2(4):1–6, 2012.

[12] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, page 145–156, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[13] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. *arXiv preprint arXiv:2212.08051*, 2022.

[14] Zheng Ding, Xuaner Zhang, Zhihao Xia, Lars Jebe, Zhuowen Tu, and Xiuming Zhang. Diffusionrig: Learning personalized priors for facial appearance editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12736–12746, 2023.

[15] Epic Games. Unreal engine. `https://www.unrealengine.com`.

[16] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. *arXiv preprint arXiv:2403.12013*, 2024.

[17] David Futschik, Kelvin Ritland, James Vecore, Sean Fanello, Sergio Orts-Escolano, Brian Curless, Daniel Sỳkora, and Rohit Pandey. Controllable light diffusion for portraits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8412–8421, 2023.

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[20] Michael Janner, Jiajun Wu, Tejas D Kulkarni, Ilker Yildirim, and Josh Tenenbaum. Self-supervised intrinsic image decomposition. *Advances in neural information processing systems*, 30, 2017.

[21] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2023.

[22] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020.

[23] 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.

[24] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[25] Hoon Kim, Minje Jang, Wonjun Yoon, Jisoo Lee, Donghyun Na, and Sanghyun Woo. Switchlight: Co-design of physics-driven architecture and pre-training framework for human portrait relighting. *arXiv preprint arXiv:2402.18848*, 2024.

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

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

[28] Peter Kocsis, Vincent Sitzmann, and Matthias Nießner. Intrinsic image diffusion for single-view material estimation. *arXiv preprint arXiv:2312.12274*, 2023.

[29] Zhengfei Kuang, Kyle Olszewski, Menglei Chai, Zeng Huang, Panos Achlioptas, and Sergey Tulyakov. Neroic: Neural rendering of objects from online image collections. *ACM Transactions on Graphics (TOG)*, 41(4):1–12, 2022.

[30] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6), nov 2017.

[31] Zhengqi Li and Noah Snavely. Learning intrinsic image decomposition from watching the world. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9039–9048, 2018.

[32] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Trans. Graph.*, 37(6), dec 2018.

[33] Xian Liu, Jian Ren, Aliaksandr Siarohin, Ivan Skorokhodov, Yanyu Li, Dahua Lin, Xihui Liu, Ziwei Liu, and Sergey Tulyakov. Hyperhuman: Hyper-realistic human generation with latent structural diffusion. *arXiv preprint arXiv:2310.08579*, 2023.

[34] Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. *ACM Transactions on Graphics (TOG)*, 42(4):1–22, 2023.

[35] Yunfei Liu, Yu Li, Shaodi You, and Feng Lu. Unsupervised learning for intrinsic image decomposition from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3248–3257, 2020.

[36] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023.

[37] 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.

[38] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022.

[39] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[40] Rohit Pandey, Sergio Orts-Escolano, Chloe Legendre, Christian Haene, Sofien Bouaziz, Christoph Rhemann, Paul E Debevec, and Sean Ryan Fanello. Total relighting: learning to relight portraits for background replacement. *ACM Trans. Graph.*, 40(4):43–1, 2021.

[41] Bui Tuong Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, jun 1975.

[42] Puntawat Ponglertnapakorn, Nontawat Tritrong, and Supasorn Suwajanakorn. Difareli: Diffusion face relighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22646–22657, 2023.

[43] Mallikarjun B R, Ayush Tewari, Abdallah Dib, Tim Weyrich, Bernd Bickel, Hans-Peter Seidel, Hanspeter Pfister, Wojciech Matusik, Louis Chevallier, Mohamed Elgharib, and Christian Theobalt. Photoapp: photorealistic appearance editing of head portraits. *ACM Trans. Graph.*, 40(4), jul 2021.

[44] 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.

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

[46] 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.

[47] 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.

[48] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[49] Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, and Daniel Aliaga. Objectstitch: Object compositing with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18310–18319, 2023.

[50] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021.

[51] Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019.

[52] Unity Technologies. Unity. `https://www.unity.com`.

[53] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, page 195–206, Goslar, DEU, 2007. Eurographics Association.

[54] Yifan Wang, Aleksander Holynski, Xiuming Zhang, and Xuaner Zhang. Sunstage: Portrait reconstruction and relighting using the sun as a light stage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20792–20802, 2023.

[55] Zhibo Wang, Xin Yu, Ming Lu, Quan Wang, Chen Qian, and Feng Xu. Single image portrait relighting via explicit multiple reflectance channel modeling. *ACM Transactions on Graphics (TOG)*, 39(6):1–13, 2020.

[56] Felix Wimbauer, Shangzhe Wu, and Christian Rupprecht. De-rendering 3d objects in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18490–18499, 2022.

[57] Liwen Wu, Rui Zhu, Mustafa B Yaldiz, Yinhao Zhu, Hong Cai, Janarbek Matai, Fatih Porikli, Tzu-Mao Li, Manmohan Chandraker, and Ravi Ramamoorthi. Factorized inverse path tracing for efficient and accurate material-lighting estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3848–3858, 2023.

[58] Shangzhe Wu, Ameesh Makadia, Jiajun Wu, Noah Snavely, Richard Tucker, and Angjoo Kanazawa. De-rendering the world's revolutionary artefacts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6338–6347, 2021.

[59] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1–10, 2020.

[60] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023.

[61] Yu-Ying Yeh, Koki Nagano, Sameh Khamis, Jan Kautz, Ming-Yu Liu, and Ting-Chun Wang. Learning to relight portrait images via a virtual light stage and synthetic-to-real adaptation. *ACM Transactions on Graphics (TOG)*, 41(6):1–21, 2022.

[62] Ye Yu and William AP Smith. Inverserendernet: Learning single image inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2019.

[63] Chong Zeng, Yue Dong, Pieter Peers, Youkang Kong, Hongzhi Wu, and Xin Tong. Dilightnet: Fine-grained lighting control for diffusion-based image generation. *arXiv preprint arXiv:2402.11929*, 2024.

[64] Jingyang Zhang, Yao Yao, Shiwei Li, Jingbo Liu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Neilf++: Inter-reflectable light fields for geometry and material estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3601–3610, 2023.

[65] Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5565–5574, 2022.

[66] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021.

[67] Longwen Zhang, Qixuan Zhang, Minye Wu, Jingyi Yu, and Lan Xu. Neural video portrait relighting in real-time via consistency modeling. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 802–812, 2021.

[68] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (ToG)*, 40(6):1–18, 2021.

[69] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18643–18652, 2022.

[70] Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W Jacobs. Deep single-image portrait relighting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7194–7202, 2019.

[71] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

A Appendix

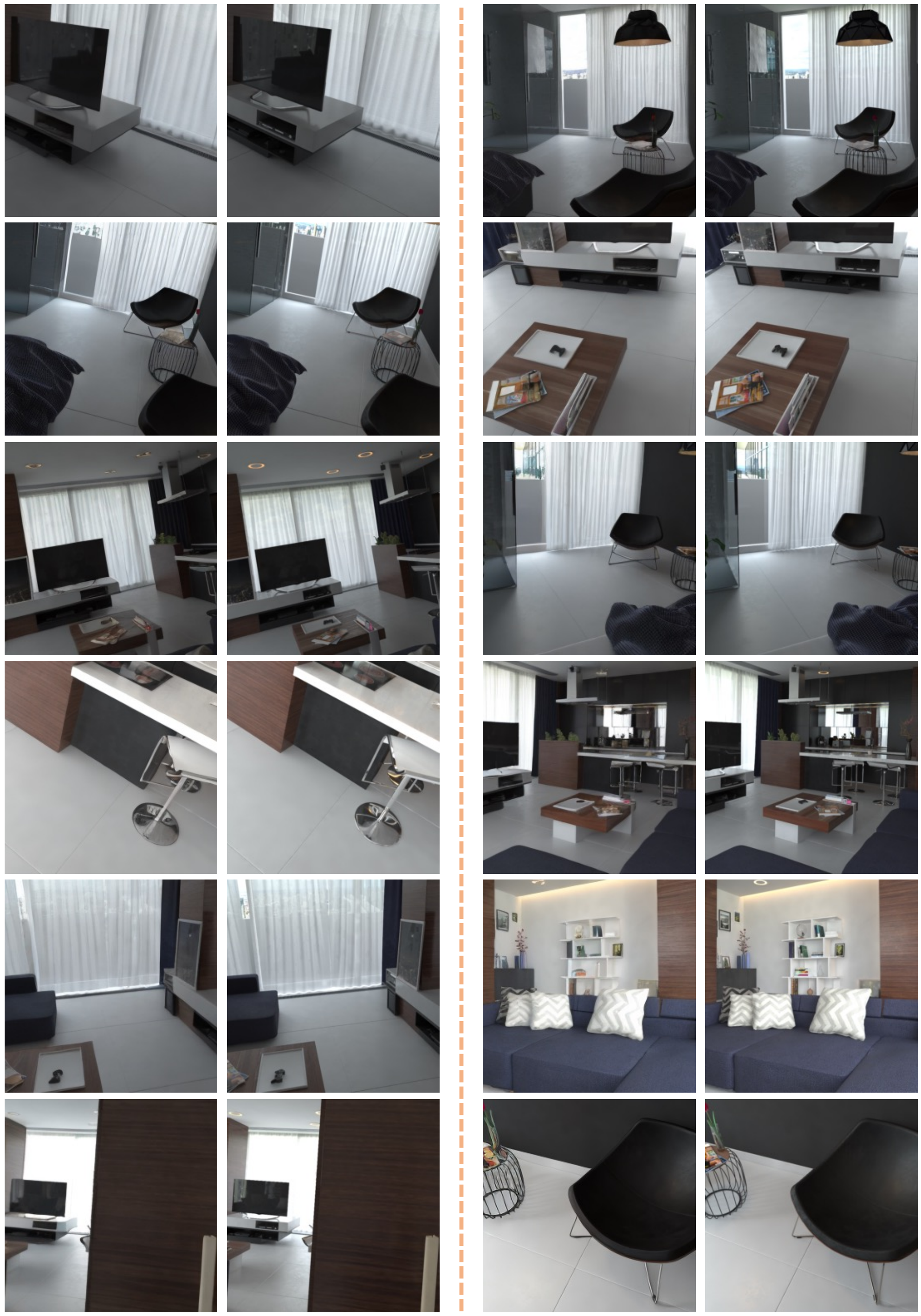

Figure 7: **Rendering results on Hypersim dataset.** The first and third columns are the rendered images while the second and the fourth are the ground truth images.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the contributions and scope of our paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Our paper discusses the limitations of our method.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our paper fully discloses all the information needed to reproduce the main experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will release our code and data soon.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All the details mentioned above are presented in the "Experiments" section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: Our paper does not include such experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This information is provided in the "Experiments" section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our paper does not discuss potential negative societal impacts of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original papers and the website that produce the training and evaluation dataset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce new assets in the "Experiments" section.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.