
LuxDiT: Lighting Estimation with Video Diffusion Transformer

Ruofan Liang^{1,2,3*} Kai He^{1,2,3} Zan Gojcic¹ Igor Gilitschenski^{2,3}
Sanja Fidler^{1,2,3} Nandita Vijaykumar^{2,3†} Zian Wang^{1,2,3†}

¹NVIDIA ²University of Toronto ³Vector Institute

Abstract

Estimating scene lighting from a single image or video remains a longstanding challenge in computer vision and graphics. Learning-based approaches are constrained by the scarcity of ground-truth HDR environment maps, which are expensive to capture and limited in diversity. While recent generative models offer strong priors for image synthesis, lighting estimation remains difficult due to its reliance on indirect visual cues, the need to infer global (non-local) context, and the recovery of high-dynamic-range outputs. We propose LuxDiT, a novel data-driven approach that fine-tunes a video diffusion transformer to generate HDR environment maps conditioned on visual input. Trained on a large synthetic dataset with diverse lighting conditions, our model learns to infer illumination from indirect visual cues and generalizes effectively to real-world scenes. To improve semantic alignment between the input and the predicted environment map, we introduce a low-rank adaptation finetuning strategy using a collected dataset of HDR panoramas. Our method produces accurate lighting predictions with realistic angular high-frequency details, outperforming existing state-of-the-art techniques in both quantitative and qualitative evaluations. Project page: <https://research.nvidia.com/labs/toronto-ai/LuxDiT/>

1 Introduction

In physically-based rendering, lighting plays a central role in shaping the appearance—how objects cast shadows, reflect, and appear integrated within a scene. From virtual object insertion and augmented reality to synthetic data generation, many downstream tasks rely on estimating scene illumination. Yet inferring lighting from casually captured images or video remains an open challenge.

A common representation of the scene illumination is the high-dynamic-range (HDR) environment map, which describes incoming light intensity from all directions. HDR maps can be acquired by using light probes or multi-exposure panoramas, requiring specialized setups that are impractical for everyday use [9]. To overcome this, several learning-based methods that estimate environment maps directly from casually captured LDR images or videos have been proposed [15, 16, 33, 74]. However, these methods typically depend on paired datasets of input images or videos and HDR environment maps, leading to a chicken-and-egg problem: a large collection of HDR environment maps is needed to train a model that aims to alleviate the need for acquiring such expensive data in the first place.

Recently, generative diffusion models have demonstrated strong capabilities in modeling complex image distributions. DiffusionLight [45] demonstrated that pretrained text-to-image models encode implicit knowledge of illumination, which can be cleverly extracted by inpainting a virtual chrome ball into an image, generating plausible appearances under varying exposure settings. However,

* Work done during internship at Nvidia

† Joint Advising



Figure 1: LuxDiT is a generative lighting estimation model that predicts high-quality HDR environment maps from visual input. It produces accurate lighting while preserving scene semantics, enabling realistic virtual object insertion under diverse conditions.

without task-specific fine-tuning, the inpainting priors of pre-trained diffusion models are insufficient for producing reliable lighting estimates in a single inference and cannot directly generate HDR outputs. As a result, DiffusionLight relies on an expensive test-time ensemble strategy to improve robustness. Moreover, sampling multiple exposures through separate inference passes introduces inconsistencies and limits the dynamic range of the reconstructed illumination.

In this work, we formulate lighting estimation as a conditional generative task and propose LuxDiT, a neural lighting predictor trained on synthetic data and adapted to real-world scenes. Conditioned on visual input, our approach fine-tunes a diffusion transformer (DiT) to synthesize HDR panoramas from noise. Unlike pixel-aligned tasks, lighting estimation requires global reasoning over scene context. DiTs are particularly suited to this task: their attention-based architecture supports global context aggregation, and their generative priors facilitate reasoning from indirect cues such as shading and reflections.

Training such a model requires diverse lighting data. To overcome the lack of real-world HDR lighting supervision, we construct a large-scale synthetic dataset with randomized geometry, materials, and lighting conditions. Training on this dataset allows the model to learn physically grounded cues for light direction and intensity. While this imparts general lighting priors, models trained purely on synthetic data often hallucinate lighting based on dataset priors, producing environment maps that are plausible but semantically mismatched with the input scene. For example, an image of an urban street may yield an environment map depicting a rural landscape. To address this, we further apply low-rank adaptation (LoRA) [24] on a curated set of real HDR panoramas, improving alignment between predicted lighting and scene semantics.

Given a single image or video, LuxDiT produces HDR environment maps with accurate direction, intensity, and scene-consistent content. It reduces lighting estimation error by 45% on Laval Outdoor sunlight direction and improves temporal consistency for video input, enabling reliable use in downstream applications such as virtual object insertion. Our main contributions are:

- A DiT-based generative architecture that synthesizes HDR environment maps from visual input.
- A LoRA-based fine-tuning strategy using curated HDR panoramas to improve semantic alignment between the input scene and predicted illumination.
- A large-scale synthetic dataset with randomized geometry, materials, and lighting.

2 Related Work

Lighting estimation aims to infer environment illumination from input imagery, and is critical for photorealistic rendering and virtual object insertion. Early learning-based methods treat lighting estimation as a supervised regression problem, predicting spherical lobes [16, 33, 72, 69], parametric sources [66, 14], or low-resolution environment maps [15, 50, 74, 52] directly from a single image. These models are trained on paired data obtained from real-world captures [15, 50, 58] or synthetic

rendering [33, 74, 52]. However, their performance often degrades in complex, in-the-wild scenes due to limited diversity in the training data.

Recent methods incorporate generative priors to address the ambiguity of scene illumination. StyleLight [55] fine-tunes a StyleGAN to generate LDR and HDR panoramas from latent codes, using GAN inversion at test time. However, its performance hinges on inversion quality and often breaks semantic alignment on out-of-domain inputs. EverLight [8] regresses a parametric lighting estimate and refines it with a GAN to add high-frequency detail, but relies on pseudo-labeled HDR data and struggles with complex or bright lighting. DiffusionLight [45] uses a diffusion model to inpaint a virtual chrome ball under multiple exposures, merging them into an HDR map. While visually plausible, this multi-stage process yields distorted panoramas and limited dynamic range.

Inverse rendering recovers scene properties such as geometry, material reflectance, and illumination from image observations. Lighting estimation is often treated as a subcomponent of this broader task, with prior work jointly estimating lighting alongside depth, normals, and albedo. Learning-based approaches [49, 33, 59] typically leverage physics-based constraints and use re-rendering losses to supervise predictions. However, these methods often assume simplified reflectance models such as Lambertian shading, which limits their ability to handle complex lighting effects.

Optimization-based methods leverage differentiable rendering [4, 71, 70, 6, 60, 42, 18, 34] to jointly optimize lighting parameters and other scene attributes through photometric losses and regularization terms. Some approaches [31] follow a decomposition-then-optimization strategy: estimating geometry and albedo first, then solving for lighting via optimization. Other works also explore priors from proxy geometry [65] or pretrained general models [38, 36, 43]. The optimization-based pipelines often require dense multi-view captures or known proxy geometry, and involve expensive test-time optimization procedures. In contrast, our method directly predicts HDR illumination in a feed-forward manner without requiring scene geometry or iterative inference.

Diffusion model priors. Diffusion models (DMs) have emerged as a powerful class of generative models in high-fidelity image [46, 2, 47, 7] and video synthesis [22, 73, 3, 63, 1]. Beyond generation, pretrained DMs have been adapted to perception tasks through task-specific finetuning on carefully curated datasets [62, 39, 19], showing strong results on spatially aligned predictions such as depth [30, 25, 29], surface normals [13, 64, 35], albedo [11, 31, 68, 35], and material properties [31, 68, 35, 43]. Adapting DMs to non-local tasks like lighting introduces new modeling challenges, as outputs such as HDR panoramas are not spatially-aligned with the input.

3 Preliminaries: Diffusion Models

Diffusion models learn to approximate a data distribution $p_{\text{data}}(\mathbf{x})$ through iterative denoising. Following DDPM [21], a forward process progressively adds Gaussian noise to a data sample $\mathbf{x}_0 \sim p_{\text{data}}$, producing a noisy version at timestep $t \in [1, T]$ as: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\bar{\alpha}_t$ defines the noise schedule. During training, a neural network μ_θ learns to reverse this process by minimizing:

$$\mathbb{E}_{\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}), t \sim p_t, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [\|\mu_\theta(\mathbf{x}_t; \mathbf{c}, t) - \mathbf{y}\|_2^2], \quad (1)$$

where $\mathbf{c}$ represents optional conditioning inputs. The denoising target $\mathbf{y}$ varies by formulation, and can be the noise ϵ [21], the v-prediction $\sqrt{\bar{\alpha}_t}\epsilon - \sqrt{1 - \bar{\alpha}_t}\mathbf{x}_0$ [48], or the clean signal $\mathbf{x}_0$ itself [28]. At inference time, samples are generated by denoising an initial Gaussian sample through a fixed number of reverse steps. In this paper, we build on CogVideoX [63], a latent video diffusion model trained on compressed video representations. A pretrained auto-encoder pair $\{\mathcal{E}, \mathcal{D}\}$ maps RGB videos to and from a latent space, such that $\mathcal{E}(\mathbf{x}) = \mathbf{z}$ and $\mathcal{D}(\mathbf{z}) \approx \mathbf{x}$. All diffusion training and generation is performed in this lower-dimensional latent space to reduce memory and computational.

4 Method

We propose LuxDiT, a diffusion-based generative framework for estimating high-dynamic-range (HDR) environment maps from a single image or video. We tailor a recent video diffusion transformer architecture [63] for lighting estimation, by jointly processing denoising targets (environment lighting) and condition tokens (LDR input images) through self-attention layers. Since a single image can be treated as a one-frame video, we refer to both inputs uniformly as input video in the remainder of this section. An overview of the architecture is shown in Figure 2. In the following sections, we describe the model design, data sources, and training procedure.

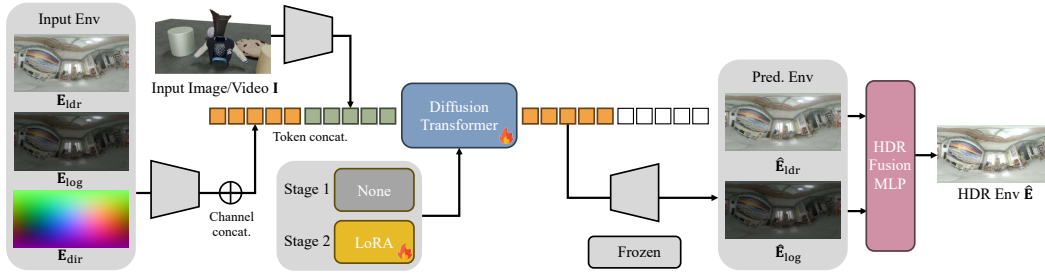


Figure 2: **Method Overview.** Given an input image or video $\mathbf{I}$, LuxDiT predicts an environment map $\mathbf{E}$ as two tone-mapped representations, $\mathbf{E}_{\text{ldr}}$ and $\mathbf{E}_{\text{log}}$, guided by a directional map $\mathbf{E}_{\text{dir}}$. Environment maps are encoded with a VAE, and the resulting latents are concatenated and jointly processed with visual input by a DiT. The outputs $\mathbf{E}_{\text{ldr}}$ and $\mathbf{E}_{\text{log}}$ are decoded and fused by a lightweight MLP to reconstruct the final HDR panorama.

4.1 Model Design

We formulate HDR environment map estimation as a conditional denoising task. Given an input video $\mathbf{I} \in \mathbb{R}^{L \times H \times W \times 3}$ with L frames, the model generates a corresponding sequence of 360° HDR panoramas $\mathbf{E} \in \mathbb{R}^{L \times H_e \times W_e \times 3}$.

Two core challenges arise: (1) standard VAEs used in latent diffusion models are trained on LDR images and cannot faithfully encode HDR content, and (2) the output panoramas are not spatially aligned with the input, requiring flexible conditioning mechanisms. We address these challenges using a dual-tonemapping HDR representation, token-based conditioning, and a unified transformer architecture that jointly denoises two latent representations of lighting.

HDR lighting representation. Realistic lighting involves high-intensity components such as the sun or artificial sources, with radiance values often exceeding 100 or 1,000. Representing this range in latent space is non-trivial: standard VAEs are trained on $[0, 1]$ -normalized LDR images and cannot reconstruct such dynamic content, and retraining on HDR data is impractical due to data scarcity.

Inspired by prior works [27, 35], we represent each HDR panorama $\mathbf{E}$ using two complementary tonemapped representations:

$$\mathbf{E}_{\text{ldr}} = \frac{\mathbf{E}}{1 + \mathbf{E}} \cdot \left(1 + \frac{\mathbf{E}}{M_{\text{ldr}}^2} \right); \quad \mathbf{E}_{\text{log}} = \frac{\log(1 + \mathbf{E})}{\log(1 + M_{\text{log}})} \quad (2)$$

where $\mathbf{E}_{\text{ldr}}$ is a standard Reinhard tonemapping and $\mathbf{E}_{\text{log}}$ captures normalized log-intensity. We set $M_{\text{ldr}} = 16$ and $M_{\text{log}} = 10,000$. Both outputs are clipped to $[0, 1]$ before VAE encoding.

At inference time, the HDR environment map is reconstructed using a lightweight MLP ψ :

$$\hat{\mathbf{E}} = \psi(\mathbf{E}_{\text{ldr}}, \mathbf{E}_{\text{log}}). \quad (3)$$

Diffusion latents. Our model builds on a transformer-based diffusion model μ_θ , adapted to predict HDR environment maps from visual input. The model operates in latent space and jointly denoises two tonemapped representations of the HDR lighting.

The tonemapped inputs $\mathbf{E}_{\text{ldr}}$ and $\mathbf{E}_{\text{log}}$ are encoded by the pretrained VAE into latent tensors $[\mathbf{z}^{\text{ldr}}, \mathbf{z}^{\text{log}}]$ with shape as $\mathbb{R}^{l \times h_e \times w_e \times C}$. These are concatenated along the channel dimension to form the diffusion target $\mathbf{z} = [\mathbf{z}^{\text{ldr}}, \mathbf{z}^{\text{log}}] \in \mathbb{R}^{l \times h_e \times w_e \times 2C}$. The input and output projection layers of the diffusion network μ_θ are extended to accommodate the increased channel dimension.

Conditioning visual input in DiT. Accurate lighting estimation requires the model to extract fine-grained shading cues from the input image, such as shadow orientation, surface reflections, and specular highlights. Unlike pixel-aligned image-to-image translation tasks, we empirically observe that concatenating conditions to the noisy latents leads to poor performance (see Table 7), indicating the need for a more flexible conditioning mechanism.

To this end, we adopt a fully attention-based architecture for the input video conditions. Specifically, we encode the input video $\mathbf{I} \in \mathbb{R}^{L \times H \times W \times 3}$ into a latent tensor $\mathcal{E}(\mathbf{I}) \in \mathbb{R}^{l \times h \times w \times C}$ using the pre-trained VAE encoder, and flatten it into a token sequence $\mathbf{c} \in \mathbb{R}^{lhw \times C}$. To help the model distinguish between condition tokens and denoising targets, we apply separate adaptive layer normalization (AdaLN) modules [44, 63] to each token type at every transformer block.

Directional embedding. To improve angular continuity in the predicted panoramas, we inject directional information into the model. Specifically, we construct a direction map of unit vectors $\mathbf{E}_{\text{dir}}$ that encodes per-pixel lighting directions in the camera coordinate system. This map is passed through the same VAE encoder $\mathcal{E}$, then projected and fused into the noise tokens using channel-wise concatenation before the transformer blocks. During training, we apply random horizontal rotations to $\mathbf{E}_{\text{dir}}$ to encourage rotational equi-variance and robust directional encoding.

Conditioned denoising process. To put it together, at each denoising timestep t , the model receives a noisy latent $\mathbf{z}_t = [\mathbf{z}_t^{\text{ldr}}, \mathbf{z}_t^{\text{log}}]$ and predicts the corresponding clean latents conditioned on visual input as $\mu_\theta(\mathbf{z}_t; \mathbf{c}, t)$. This transformer-based design allows the model to propagate indirect lighting cues—such as shadows and reflections—through global self-attention, enabling lighting prediction that is both scene-consistent and directionally accurate.

4.2 Data Strategy

Supervised training of our model requires paired data in the form $(\mathbf{I}, \mathbf{E}_{\text{ldr}}, \mathbf{E}_{\text{log}})$, where $\mathbf{I}$ is an LDR input and $\mathbf{E}_{\text{ldr}}, \mathbf{E}_{\text{log}}$ are tonemapped versions of the target HDR environment map. To overcome the scarcity of real-world HDR annotations, we leverage three complementary data sources: synthetic renderings, HDR panorama images, and LDR panoramic videos.

Synthetic rendering data. To supervise lighting prediction using physically accurate visual cues, we generate synthetic data by rendering randomized 3D scenes lit by HDR environment maps. Each scene consists of (i) a ground plane with randomly assigned PBR materials, (ii) 3D objects sampled from Objaverse [10], and (iii) simple geometric primitives such as spheres, cubes, and cylinders with varied materials. We render multiple frames per scene with randomized camera trajectories and environment map rotations. Despite their simplicity, these scenes exhibit diverse lighting effects, including cast shadows, specular highlights, and inter-reflections, all paired with ground-truth HDR illumination. Empirically, we find this data is critical for enabling the model to learn accurate shading cues and light-source location (see Table 7).

HDR panorama images. We generate training pairs by sampling perspective crops from HDR environment maps with data augmentation. Specifically, given a panorama, we randomly sample camera parameters including azimuth, elevation, field of view, and exposure scale. These parameters define a virtual pinhole camera, which we use to project the panorama into an LDR perspective view $\mathbf{I}$. The corresponding HDR environment map serves as the ground truth lighting target $\mathbf{E}$. To support temporal training, we extend this procedure to generate multi-frame sequences by smoothly varying the camera pose over time.

LDR panorama videos. To enable the generation of dynamic panorama environment maps, we also incorporate training data from LDR panoramic videos. Although ground-truth HDR environment maps are not available for this source, we use it in the form $(\mathbf{I}, \mathbf{E}_{\text{ldr}}, \emptyset)$, where $\mathbf{E}_{\text{ldr}}$ is derived using tonemapping and $\emptyset$ indicates the absence of log-space intensity. The panoramic video is projected into a perspective-view video using randomized camera parameters, following the same procedure as above. Despite the lack of HDR intensity, this data improves robustness and temporal consistency by exposing the model to natural image statistics, motion patterns, and diverse real-world lighting conditions. We use 2,000 panoramic videos from the WEB360 dataset [57] for training, and hold out 114 videos for evaluation.

4.3 Training Scheme

We adopt a two-stage training strategy to progressively build the model’s capacity and improve generalization. The first stage focuses on learning physically grounded lighting cues from synthetic data. The second stage adapts the model to real-world distributions through LoRA-based fine-tuning.

Stage I: Synthetic supervised training. We begin by training the model on the synthetic rendering dataset described in Section 4.2. This stage enables the model to learn the fundamental relationship between image-based shading cues and HDR environment lighting.

We follow the standard DDPM training objective [21] adopted by the CogVideoX base model [63]:

$$\mathcal{L}_1(\theta) = \mathbb{E}_{\mathbf{z}_0, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(T)} [\|\epsilon - \mu_\theta(\mathbf{z}_t, \mathbf{c}, t)\|_2^2], \quad (4)$$

where $\mathbf{z}_0$ denotes the clean latent pair $[\mathbf{z}^{\text{ldr}}, \mathbf{z}^{\text{log}}]$, and $\mathbf{c}$ is the conditioning latent from the input video. During training, we randomly drop either $\mathbf{z}^{\text{ldr}}$ or $\mathbf{z}^{\text{log}}$ with probability $p = 0.1$ to encourage robustness to missing tonemapped representations.

Table 1: Comparison of our method with baselines on three benchmark datasets. The results are reported in terms of scale-invariant RMSE, angular error, and normalized RMSE.

Dataset	Method	Scale-invariant RMSE ↓			Angular Error ↓			Normalized RMSE ↓		
		Diffuse	Matte	Mirror	Diffuse	Matte	Mirror	Diffuse	Matte	Mirror
Laval Indoor	StyleLight	0.135	0.315	0.552	4.238	4.742	6.781	0.234	0.404	0.511
	DiffusionLight	0.124	0.325	0.597	2.500	3.421	5.936	0.216	0.361	0.431
	Ours	0.112	0.297	0.586	2.555	3.526	5.641	0.196	0.341	0.457
Laval Outdoor	H-G et al. [23]	0.300	0.437	0.587	7.851	8.755	26.052	0.551	0.627	0.740
	NLFE	0.112	0.234	0.431	4.804	5.279	7.278	0.217	0.331	0.496
	DiffusionLight	0.083	0.224	0.414	1.936	2.955	5.491	0.167	0.330	0.472
	Ours	0.068	0.190	0.396	2.018	2.939	5.286	0.137	0.271	0.454
Poly Haven	StyleLight	0.138	0.336	0.620	3.034	4.272	6.602	0.198	0.344	0.474
	NLFE	0.159	0.326	0.571	3.305	4.240	5.180	0.224	0.365	0.458
	DiffusionLight	0.113	0.270	0.519	2.199	3.121	4.104	0.191	0.282	0.391
	Ours	0.077	0.196	0.442	1.235	1.977	2.783	0.111	0.199	0.323

Table 2: Angular error on estimated peak luminance light direction on Laval Outdoor sunny scenes.

Method	Peak Angular Error ↓	
	Mean	Median
H-G et al. [23]	52.8	47.8
NLFE	52.9	43.5
DiffusionLight	44.4	32.1
Ours	23.7	17.5

Table 3: Quantitative comparison with video input. Peak angular error (PAE) is used to evaluate PolyHaven-Peak videos. Angular error (AE) on is used to evaluate WEB360 LDR videos.

Method	PolyHaven-Peak		WEB360	
	PAE Mean ↓	PAE Std ↓	AE ↓	AE Std ↓
DiffusionLight	19.09	10.31	6.504	0.269
Ours (image)	5.74	3.68	5.679	0.382
Ours (video)	5.21	1.95	5.218	0.072

Stage II: Semantic adaptation. After base training, we fine-tune the model to improve semantic alignment between the input appearance and the predicted HDR environment map.

This stage uses real-world data sources, including perspective projections from HDR panoramas and LDR panoramic videos. Since HDR ground truth is not available in the latter, we supervise only the LDR-tonemapped component. To avoid overfitting and preserve the pretrained model capacity, we apply parameter-efficient LoRA fine-tuning [24], optimizing a small set of injected low-rank parameters $\Delta\theta$ in the transformer layers:

$$\mathcal{L}_{\Pi}(\Delta\theta) = \mathbb{E}_{\mathbf{z}_0, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(T)} [\|\epsilon - \boldsymbol{\mu}_{\theta + \Delta\theta}(\mathbf{z}_t, \mathbf{c}, t)\|_2^2], \quad (5)$$

5 Experiments

5.1 Experiment Settings

Implementation details. We use the pre-trained CogVideoX [63] model as our backbone. All training is conducted on 16 NVIDIA A100 GPUs. Input resolutions are randomly sampled between 512×512 and 480×720 , and output environment map resolutions are between 128×256 and 256×512 . The image-based model is trained with a batch size of 192 for 12,000 iterations. For video training, we use the same spatial resolutions and uniformly sample frame lengths from 9, 17, 25. The video model is trained with an average batch size of 48 for an additional 12,000 iterations. LoRA modules are applied to all attention layers with a rank of 64. We fine-tune the LoRA parameters for 5,000 iterations during the adaptation stage. Please refer to supplement for implementation details.

Datasets. We evaluate our method on the following three benchmark datasets, covering various indoor and outdoor scenes. 1) Laval Indoor [15]: We use the same set of 289 test HDRIs used by prior works [45, 55]; 2) Laval Outdoor [23]: We evaluate on 116 sunny HDR panoramas with concentrated sunlight selected from the original dataset; 3) Poly Haven [67]: We select 181 Poly Haven HDRIs not used during model training to evaluate performance across both indoor and outdoor scenes.

Metrics. Following prior works [55, 45], we use three standard metrics for evaluating HDR lighting: scale-invariant root mean square error (si-RMSE) [17], angular error in degrees [32], and normalized RMSE (n-RMSE) [45]. For scenes with concentrated sunlight, we additionally report peak angular error (PAE) [23, 58], which measures the angular deviation of the predicted peak light direction.

Baselines. For indoor scenes, we compare against DiffusionLight [45], StyleLight [55], Weber et al. [61], and EMLight [69], using metrics reported by [45] when applicable. For outdoor scenes, we compare against DiffusionLight [45], Hold-Geoffroy *et al.* [23], and NLFE [58].

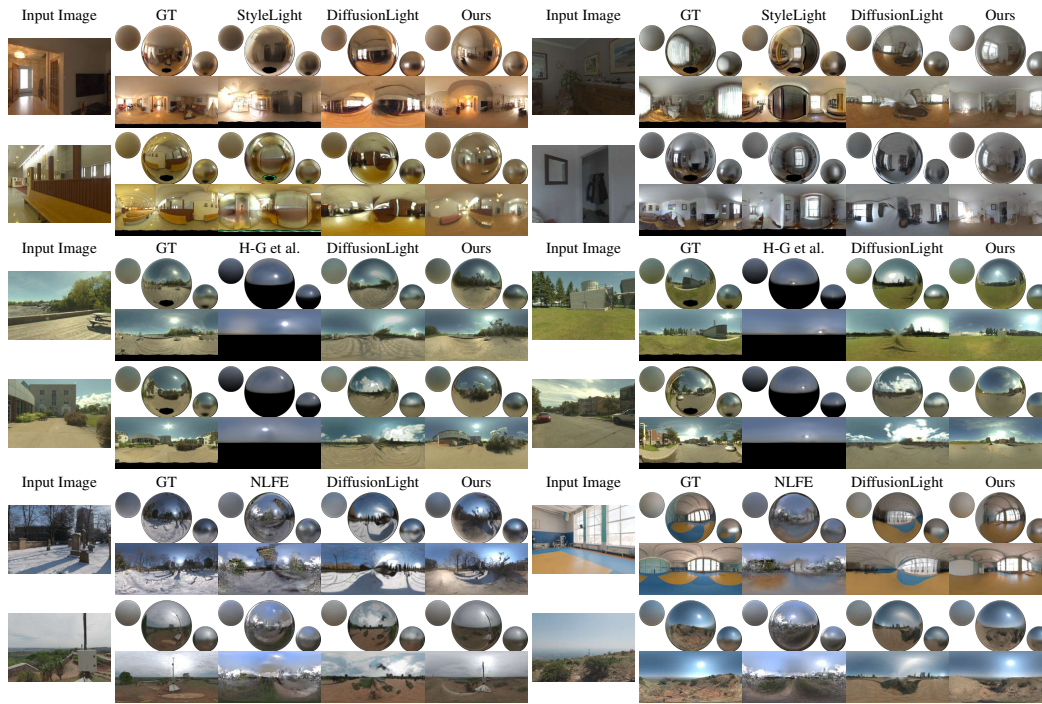


Figure 3: Qualitative comparison with baseline methods on three benchmark datasets.

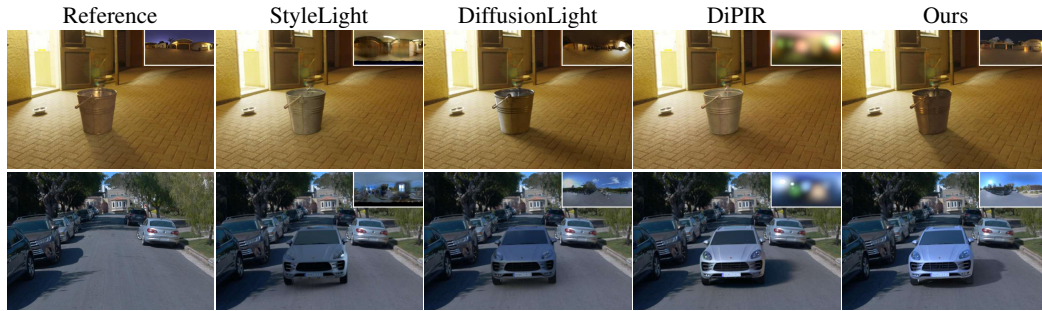


Figure 4: Qualitative comparison of virtual object insertion.

Table 4: Ablation study on impact of LoRA scale at inference time.

LoRA Scale	Diffuse ↓	Matte ↓	Mirror ↓
0.00	2.98	5.02	6.07
0.25	2.09	3.69	4.67
0.50	1.52	2.66	3.56
0.75	1.22	2.05	2.88
1.00	1.17	1.92	2.72

Table 5: Ablation study on impact of camera field-of-view.

FOV	Diffuse ↓	Matte ↓	Mirror ↓
45°	1.29	2.14	2.95
50°	1.26	2.06	2.86
60°	1.17	1.92	2.72
70°	1.15	1.85	2.63
75°	1.13	1.80	2.59

Table 6: Ablation study on impact of camera elevation.

Elevation	Diffuse ↓	Matte ↓	Mirror ↓
-30°	1.70	3.04	3.95
-15°	1.22	2.05	2.87
0°	1.17	1.92	2.72
15°	1.28	2.09	2.94
30°	1.71	2.59	3.51

Table 7: Ablation study on model design choices and training data. We report the angular error with three-spheres protocol.

Settings	Laval Indoor			Poly Haven		
	Diffuse ↓	Matte ↓	Mirror ↓	Diffuse ↓	Matte ↓	Mirror ↓
Ours (channel concat.)	7.09	10.04	11.07	7.09	10.04	11.07
Ours (w/o synthetic data)	4.50	5.14	6.96	1.48	2.08	2.86
Ours	2.56	3.53	5.64	1.23	1.98	2.78

5.2 Evaluation of Image Lighting Estimation

We follow the evaluation protocol from prior work to render spheres with three representative materials (gray-diffuse, silver-matte, and mirror), using the estimated HDR environment map from the LDR input image [15, 55, 45].

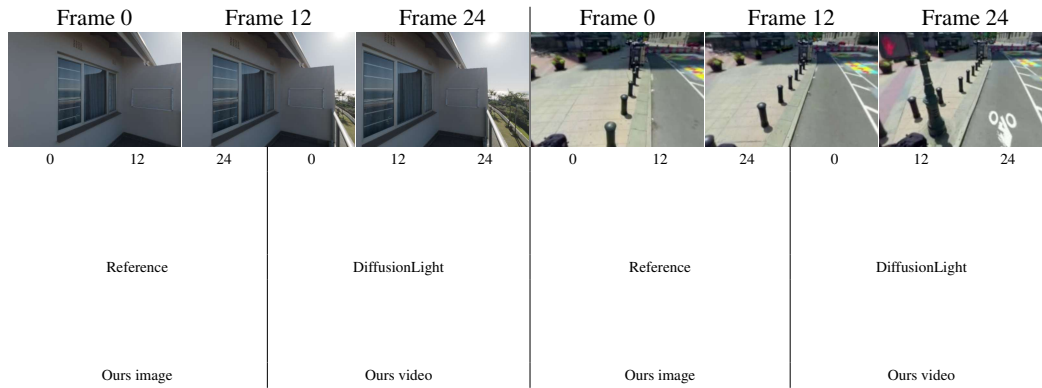


Figure 5: Qualitative comparison of video lighting estimation.

Table 1 reports quantitative comparisons on three benchmarks spanning both indoor and outdoor scenes. On the Laval Indoor dataset, our method performs comparably or better than DiffusionLight across most metrics, despite not using Laval Indoor dataset during training. This dataset exhibits a noticeable shift in color and intensity distribution compared to our training set, and our strong performance demonstrates robust generalization.

From qualitative comparison shown in Figure 3, DiffusionLight can lose angular high-frequency details from the input image due to its distorted representation. In contrast, our estimated environment maps can recover more high-frequency details while preserving accurate lighting.

On the Laval Outdoor and Poly Haven datasets with a broader dynamic range, our method consistently outperforms prior state-of-the-art methods. Hold-Geoffroy *et al.* [23] can estimate concentrated peak light source such as sunlight; however, its results do not adapt well to the details of the input image. NLFE [58] can estimate in-context environment maps, but it often fails to estimate accurate highlights. DiffusionLight performs better than other baselines, but due to its limited dynamic range, it struggles with outdoor high-intensity light sources.

To further assess directional accuracy, we evaluate the angular error of the peak luminance direction on a subset of the Laval Outdoor dataset containing direct sunlight. Table 2 reports the mean and median peak angular errors. Our method reduces peak angular error by nearly 50% compared to DiffusionLight, confirming its advantage in capturing accurate light direction—a critical factor for casting realistic shadows in downstream applications such as object insertion.

5.3 Evaluation of Video Lighting Estimation

To evaluate lighting estimation accuracy and consistency on video input, we construct two types of test sequences:

- *PolyHaven-Peak*: We project 12 unseen Poly Haven panoramas (each with direct sunlight) into videos using a smooth panning camera. This setting is used to evaluate peak angular error.
- *WEB360*: We randomly select 12 LDR panoramic videos featuring dynamic content from WEB360 and render them into perspective views with fixed horizontal camera motion. This setting evaluates temporal consistency using chromatic angular error on rendered mirror spheres.

Each set contains 12 videos at resolution of 480×720 and a length of 25 frames. To quantify temporal consistency, we compute the standard deviation (std) of per-frame error metrics for each video clip, and average the results across the 12-video set.

We compare our video inference to two baselines: our own image-based inference (applied frame-by-frame) and DiffusionLight [45]. Table 3 reports the results. Our method outperforms DiffusionLight. Comparing to Ours (image), video inference achieves higher accuracy and significantly lower temporal variance, indicating more stable predictions across time.

Figure 5 shows qualitative examples of video inference. Both DiffusionLight and our image-based variant exhibit visible temporal flickering. In contrast, our method produces smooth lighting transitions, successfully aligning content across frames and preserving consistent lighting behavior over time.

5.4 Evaluation of Virtual Object Insertion

Virtual object insertion is a key downstream application of lighting estimation. We evaluate our method on this task using the benchmark from [36], using 11 HDR panoramas from the Poly Haven dataset [67]. For each scene, a virtual object and a known ground plane are manually placed into the environment. Each test case includes an LDR background image rendered from the HDR panorama, along with a posed object and ground plane. A pseudo-ground-truth object insertion is generated by rendering the object using the original HDR environment map. This allows for controlled comparison against renderings produced using predicted lighting.

We report quantitative metrics in Table 8. In addition, we conduct a user study to assess perceptual quality (details provided in the supplement), and report the percentage of samples where users preferred our results over baseline methods.

Our method achieves visual quality comparable to DiPIR and significantly outperforms other baselines. Notably, DiPIR is specialized for object insertion and incorporates additional modules for tone mapping and appearance harmonization. In contrast, our model estimates lighting alone, yet still produces realistic composite renderings. We include qualitative results in Figure 4.

5.5 Ablation Study

Model Design and Training Data We evaluate two model variants to ablate the contributions of our architectural and training design: (1) *Channel concatenation*: This variant fuses input and environment map (resized to match input image) latents along the channel dimension [27], and no token-wise concatenation is used. Our two-stage training is also applied. (2) *Training without synthetic data*: This variant skips Stage I training and uses only panorama crops for fine-tuning.

Table 7 reports angular errors on Laval Indoor and Poly Haven. Channel concatenation significantly underperforms, confirming the importance of token-level conditioning. Without synthetic pretraining, the model performs well in-domain (Poly Haven) but degrades out-of-domain (Laval Indoor), showing synthetic data pre-training is crucial for learning generalized lighting priors.

LoRA scale. We vary the LoRA interpolation weight from 0.0 to 1.0 to ablate how fine-tuned LoRA affects the predicted lighting content. Table 4 shows that higher LoRA weights yield lower angular error on Poly Haven, validating the effectiveness of LoRA for improving semantic alignment.

Camera sensitivity. We test robustness to camera variation by rendering crops from Poly Haven under varying field of view (45° to 75°) and camera elevation (-30° to 30°). Results in Tables 5 and 6 show that while extreme viewpoints introduce mild error increases, performance remains stable, demonstrating robustness to moderate viewpoint shifts.

6 Discussion

We introduce LuxDiT, a conditional generative model for estimating HDR scene illumination from casually captured images and videos. Our approach fine-tunes a video diffusion transformer (DiT) to synthesize HDR environment maps, combining large-scale synthetic data for learning physically grounded priors with LoRA-based adaptation on real HDR panoramas to improve semantic alignment. Extensive experiments demonstrate that LuxDiT produces accurate, high-frequency, and scene-consistent lighting predictions from limited visual input.

Limitations and future work. While LuxDiT produces high-quality lighting predictions, inference remains computationally intensive due to the iterative nature of diffusion models, limiting its use in real-time applications. Future work could explore model distillation or more efficient architectures to accelerate inference. Additionally, the resolution of predicted panoramas is limited by data and training scale; generating high-resolution outputs for immersive applications will require richer, more diverse HDR supervision. Looking ahead, with recent progress in joint generative modeling [5, 37], we see LuxDiT as a step toward unified inverse and forward rendering frameworks, complementing recent progress in neural forward rendering and G-buffer estimation [68, 35, 20]. Future directions include joint modeling or co-training of lighting, geometry, and material for general-purpose scene reconstruction and appearance synthesis.

Table 8: Quantitative evaluation of virtual object insertion. We report the percentage of images where users preferred Ours over baselines. A preference $> 50\%$ indicates Ours outperforming baselines.

Method	RMSE $\downarrow$	SSIM $\uparrow$	Ours Preferred
StyleLight	0.056	0.986	60.6%
DiffusionLight	0.057	0.987	60.6%
DiPIR	0.048	0.989	54.5%
Ours	0.047	0.990	/

Acknowledgments and Disclosure of Funding

The authors thank Jacob Munkberg, Jon Hasselgren, Jun Gao, and Huan Ling for their insightful discussions. This work was conducted at and supported by NVIDIA.

References

- [1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. [arXiv preprint arXiv:2501.03575](#), 2025.
- [2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. eDiff-I: text-to-image diffusion models with ensemble of expert denoisers. [arXiv preprint arXiv:2211.01324](#), 2022.
- [3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In [IEEE Conference on Computer Vision and Pattern Recognition \(CVPR\)](#), 2023.
- [4] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. NeRD: neural reflectance decomposition from image collections. In [ICCV](#), 2021.
- [5] Hila Chefer, Uriel Singer, Amit Zohar, Yuval Kirstain, Adam Polyak, Yaniv Taigman, Lior Wolf, and Shelly Sheynin. VideoJAM: Joint appearance-motion representations for enhanced motion generation in video models. [arXiv: 2502.02492](#), 2025.
- [6] Wenzheng Chen, Joey Litalien, Jun Gao, Zian Wang, Clement Fuji Tsang, Sameh Khalis, Or Litany, and Sanja Fidler. DIB-R++: Learning to predict lighting and material with a hybrid differentiable renderer. In [NeurIPS](#), 2021.
- [7] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. [arXiv preprint arXiv:2309.15807](#), 2023.
- [8] Mohammad Reza Karimi Dastjerdi, Jonathan Eisenmann, Yannick Hold-Geoffroy, and Jean-François Lalonde. Everlight: Indoor-outdoor editable hdr lighting estimation. In [Proceedings of the IEEE/CVF International Conference on Computer Vision \(ICCV\)](#), pages 7420–7429, October 2023.
- [9] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In [Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '97](#), page 369–378, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [10] 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. In [Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition](#), pages 13142–13153, 2023.
- [11] Xiaodan Du, Nicholas Kolkin, Greg Shakhnarovich, and Anand Bhattad. Generative models: What do they know? do they know things? let’s find out!, 2024.
- [12] Egor Ershov, Alexey Savchik, Illya Semenov, Nikola Banić, Alexander Belokopytov, Daria Senshina, Karlo Koščević, Marko Subašić, and Sven Lončarić. The cube++ illumination estimation dataset. [IEEE access](#), 8:227511–227527, 2020.
- [13] 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. In [ECCV](#), 2024.
- [14] Marc-André Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagné, and Jean-François Lalonde. Deep parametric indoor lighting estimation. In [ICCV](#), pages 7175–7183, 2019.
- [15] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. [arXiv preprint arXiv:1704.00090](#), 2017.
- [16] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-François Lalonde. Fast spatially-varying indoor lighting estimation. In [CVPR](#), pages 6908–6917, 2019.

- [17] Roger Grosse, Micah K. Johnson, Edward H. Adelson, and William T. Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*, pages 2335–2342. IEEE, 2009.
- [18] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, light, and material decomposition from images using Monte Carlo rendering and denoising. *arXiv:2206.03380*, 2022.
- [19] Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Zhang, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. *arXiv preprint arXiv:2409.18124*, 2024.
- [20] 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.
- [21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [22] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022.
- [23] Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-François Lalonde. Deep sky modeling for single image outdoor lighting estimation. In *CVPR*, pages 6927–6935, 2019.
- [24] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.
- [25] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. In *CVPR*, 2025.
- [26] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. Mitsuba 3 renderer, 2022. <https://mitsuba-renderer.org>.
- [27] Haian Jin, Yuan Li, Fujun Luan, Yuanbo Xiangli, Sai Bi, Kai Zhang, Zexiang Xu, Jin Sun, and Noah Snavely. Neural gaffer: Relighting any object via diffusion. In *Advances in Neural Information Processing Systems*, 2024.
- [28] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022.
- [29] Bingxin Ke, Dominik Narnhofer, Shengyu Huang, Lei Ke, Torben Peters, Katerina Fragkiadaki, Anton Obukhov, and Konrad Schindler. Video depth without video models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- [30] 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.
- [31] Peter Kocsis, Vincent Sitzmann, and Matthias Nießner. Intrinsic image diffusion for single-view material estimation. In *arxiv*, 2023.
- [32] Chloe LeGendre, Wan-Chun Ma, Graham Fyffe, John Flynn, Laurent Charbonnel, Jay Busch, and Paul Debevec. Deeplight: Learning illumination for unconstrained mobile mixed reality. In *CVPR*, pages 5918–5928, 2019.
- [33] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *CVPR*, pages 2475–2484, 2020.
- [34] Ruofan Liang, Huiting Chen, Chunlin Li, Fan Chen, Selvakumar Panneer, and Nandita Vijaykumar. Envird: Implicit differentiable renderer with neural environment lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 79–89, 2023.
- [35] 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.

- [36] Ruofan Liang, Zan Gojcic, Merlin Nimier-David, David Acuna, Nandita Vijaykumar, Sanja Fidler, and Zian Wang. Photorealistic object insertion with diffusion-guided inverse rendering. In ECCV, 2024.
- [37] Yuanxun Lu, Jingyang Zhang, Tian Fang, Jean-Daniel Nahmias, Yanghai Tsin, Long Quan, Xun Cao, Yao Yao, and Shiwei Li. Matrix3D: Large Photogrammetry Model All-in-One, 2025.
- [38] Linjie Lyu, Ayush Tewari, Marc Habermann, Shunsuke Saito, Michael Zollhöfer, Thomas Leimkühler, and Christian Theobalt. Diffusion posterior illumination for ambiguity-aware inverse rendering. ACM Transactions on Graphics, 42(6), 2023.
- [39] Gonzalo Martin Garcia, Karim Abou Zeid, Christian Schmidt, Daan de Geus, Alexander Hermans, and Bastian Leibe. Fine-tuning image-conditional diffusion models is easier than you think. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2025.
- [40] Oscar Michel, Anand Bhattad, Eli VanderBilt, Ranjay Krishna, Aniruddha Kembhavi, and Tanmay Gupta. Object 3dit: Language-guided 3d-aware image editing. Advances in Neural Information Processing Systems, 36:3497–3516, 2023.
- [41] 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. arXiv preprint arXiv:2003.08934, 2020.
- [42] 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. arXiv:2111.12503, 2021.
- [43] Jacob Munkberg, Zian Wang, Ruofan Liang, Tianchang Shen, and Jon Hasselgren. VideoMat: Extracting PBR Materials from Video Diffusion Models. In Eurographics Symposium on Rendering - CGF Track, 2025.
- [44] William Peebles and Saining Xie. Scalable diffusion models with transformers. In Proceedings of the IEEE/CVF international conference on computer vision, pages 4195–4205, 2023.
- [45] Pakkapon Phongthawee, Worameth Chinchuthakun, Nontaphat Sinsunthithet, Amit Raj, Varun Jampani, Pramook Khungurn, and Supasorn Suwajanakorn. DiffusionLight: light probes for free by painting a chrome ball. In ArXiv, 2023.
- [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 (CVPR), 2022.
- [47] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. Advances in Neural Information Processing Systems, 35:36479–36494, 2022.
- [48] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net, 2022.
- [49] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W. Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In ICCV, 2019.
- [50] Shuran Song and Thomas Funkhouser. Neural illumination: Lighting prediction for indoor environments. In CVPR, pages 6918–6926, 2019.
- [51] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 2149–2159, 2022.
- [52] Jiajun Tang, Yongjie Zhu, Haoyu Wang, Jun-Hoong Chan, Si Li, and Boxin Shi. Estimating spatially-varying lighting in urban scenes with disentangled representation. In ECCV, 2022.
- [53] Benjamin Ummerhofer, Sanskar Agrawal, Rene Sepulveda, Yixing Lao, Kai Zhang, Tianhang Cheng, Stephan Richter, Shenlong Wang, and German Ros. Objects with lighting: A real-world dataset for evaluating reconstruction and rendering for object relighting. In 2024 International Conference on 3D Vision (3DV), pages 137–147. IEEE, 2024.

- [54] Giuseppe Vecchio and Valentin Deschaintre. Matsynth: A modern pbr materials dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024.
- [55] Guangcong Wang, Yinuo Yang, Chen Change Loy, and Ziwei Liu. Stylelight: Hdr panorama generation for lighting estimation and editing. In European Conference on Computer Vision (ECCV), 2022.
- [56] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. NeurIPS, 2021.
- [57] Qian Wang, Weiqi Li, Chong Mou, Xinhua Cheng, and Jian Zhang. 360dvd: Controllable panorama video generation with 360-degree video diffusion model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6913–6923, 2024.
- [58] Zian Wang, Wenzheng Chen, David Acuna, Jan Kautz, and Sanja Fidler. Neural light field estimation for street scenes with differentiable virtual object insertion. In ECCV, 2022.
- [59] Zian Wang, Jonah Philion, Sanja Fidler, and Jan Kautz. Learning indoor inverse rendering with 3D spatially-varying lighting. In ICCV, 2021.
- [60] Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2023.
- [61] Henrique Weber, Mathieu Garon, and Jean-François Lalonde. Editable indoor lighting estimation. In European Conference on Computer Vision, pages 677–692. Springer, 2022.
- [62] Guangkai Xu, Yongtao Ge, Mingyu Liu, Chengxiang Fan, Kangyang Xie, Zhiyue Zhao, Hao Chen, and Chunhua Shen. What matters when repurposing diffusion models for general dense perception tasks? arXiv preprint arXiv:2403.06090, 2024.
- [63] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072, 2024.
- [64] Chongjie Ye, Lingteng Qiu, Xiaodong Gu, Qi Zuo, Yushuang Wu, Zilong Dong, Liefeng Bo, Yuliang Xiu, and Xiaoguang Han. Stablenormal: Reducing diffusion variance for stable and sharp normal. ACM Transactions on Graphics, 2024.
- [65] Hong-Xing Yu, Samir Agarwala, Charles Herrmann, Richard Szeliski, Noah Snavely, Jiajun Wu, and Deqing Sun. Accidental light probes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12521–12530, 2023.
- [66] Ye Yu and William A. P. Smith. InverseRenderNet: learning single image inverse rendering. In CVPR, 2019.
- [67] Greg Zaal and et al. Poly Haven - The Public 3D Asset Library, 2025.
- [68] Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. RGB $\leftrightarrow$ X: image decomposition and synthesis using material-and lighting-aware diffusion models. In ACM SIGGRAPH 2024 Conference Papers, pages 1–11, 2024.
- [69] Fangneng Zhan, Changgong Zhang, Yingchen Yu, Yuan Chang, Shijian Lu, Feiying Ma, and Xuansong Xie. Emlight: Lighting estimation via spherical distribution approximation. In Proceedings of the AAAI Conference on Artificial Intelligence, 2021.
- [70] 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 CVPR, 2021.
- [71] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In CVPR, 2022.
- [72] Yiqin Zhao and Tian Guo. Pointar: Efficient lighting estimation for mobile augmented reality. arXiv preprint arXiv:2004.00006, 2020.
- [73] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. arXiv preprint arXiv:2211.11018, 2022.
- [74] Yongjie Zhu, Yinda Zhang, Si Li, and Boxin Shi. Spatially-varying outdoor lighting estimation from intrinsics. In CVPR, 2021.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”,**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

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 abstract and introduction clearly state our contributions, which are fully supported by the experiments and analysis presented.

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: We discuss limitations in Section 6.

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: The paper does not contain theoretical results or formal proofs.

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: We include all necessary details for reproduction in the main paper and supplementary materials.

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: [No]

Justification: We plan to release the code and data upon acceptance. The internal guidelines of our institution prevent us from releasing code at this stage.

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 relevant details are provided in the paper and supplementary materials.

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: [No]

Justification: Due to the high computational cost of training large-scale video diffusion models, we did not perform multiple runs to report error bars.

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: We include sufficient information in supplementary materials.

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: Our work adheres to 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: [Yes]

Justification: We discussed societal impacts of our work in the supplementary material.

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 work does not involve models or data with high risk of misuse.

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: Third-party assets are properly credited, and their licenses are respected.

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: [\[NA\]](#)

Justification: Our paper does not release new assets.

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: [Yes]

Justification: We include the details of all crowdsourcing experiments in supplementary materials.

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: [Yes]

Justification: An approval is obtained for all of our experiments.

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.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLMs were not used in the core methods of this research.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

In the supplementary material, we discuss the broader impact of our project in Sec. A, and provide additional details for implementation and experiments in Sec. B. Sec. C provides additional quantitative and qualitative results. We refer to the [accompanying video](#) for extended comparisons on video lighting estimation.

A Broader Impact

We introduce LuxDiT, a generative model for estimating high-dynamic-range (HDR) environment lighting from casually captured images and videos. Lighting estimation is a core challenge in photorealistic rendering due to its non-local and indirect nature. LuxDiT produces scene-consistent HDR panoramas, enabling applications in virtual object insertion, relighting, AR/VR, and visual effects. It can also support synthetic data generation for downstream tasks in robotics and perception, where realistic illumination is critical.

Similar to other generative methods, LuxDiT could be misused to produce visually convincing but deceptive content. While it does not directly generate synthetic scenes, it enables realistic virtual object insertion and may facilitate the creation of manipulated imagery that is difficult to distinguish from real footage. We encourage responsible use of LuxDiT and caution against its deployment in contexts where synthetic content could mislead viewers or undermine public trust, such as misinformation or falsified media.

B Additional Details

B.1 HDR Reconstruction

Section 4.1 describes our method for reconstructing HDR environment maps from two tone-mapped LDR images using a lightweight MLP $\psi(\mathbf{E}_{\text{ldr}}, \mathbf{E}_{\text{log}})$. This MLP consists of 5 layers with 64 hidden units per layer and LeakyReLU activation. A softplus activation is applied to the final output layer to ensure non-negative outputs.

The MLP ψ operates on a per-pixel basis: it takes a pair of LDR RGB values as input and predicts a single HDR RGB value. It is trained using the same HDR environment maps as the diffusion model, with augmentations including random intensity rescaling and exposure adjustments for diversity. To simulate limited input precision, LDR inputs are randomly quantized to 8-bit RGB values. We train the MLP using a Huber loss with $\delta = 1.0$, which provides robustness against large HDR outliers while preserving smooth gradients.

Additionally, we show the tone-mapping curves used to generate the LDR images in Fig. 6. Our dual-tone mapping strategy ensures sufficient sampling across the full dynamic range $[0, 10,000]$, supporting accurate HDR reconstruction.

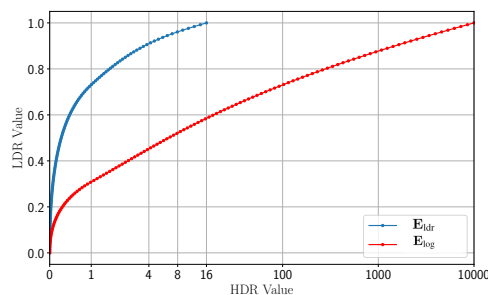


Figure 6: The two tone-mapping curves used to generate the LDR images. The 128 dot points along the curve are evenly spaced along $[0, 1]$ LDR value range.

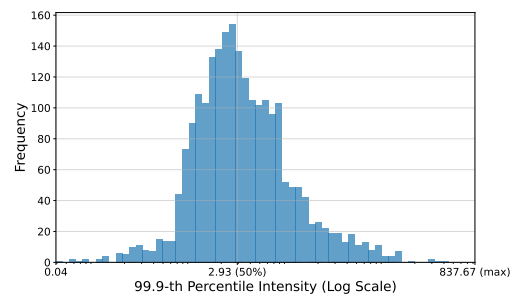


Figure 7: The histogram of the 99.9-th percentile intensity of all HDR environment maps in our training set.

B.2 Datasets

We provide more details about the datasets used in our experiments.

The data sources of HDR environment maps. We collected 2386 HDR environment maps from the following 4 data sources either publicly available or commercially available.

- *Poly Haven*⁰: 626 HDR environment maps with a wide range of indoor and outdoor lighting.
- *HDR Maps*¹: 403 HDR environment maps with diverse lighting conditions, including 294 panorama maps and 109 hemi-sphere sky maps.
- *HDRI Skies*²: 457 HDR environment maps with outdoor lighting conditions.
- *DOSCH DESIGN*³: 900 HDR environment maps mainly for outdoor lighting conditions.

Figure 7 shows the histogram of the 99.9-th percentile intensity of all HDR environment maps in our training set. With over 50% of the HDR environment maps having a 99.9-th percentile intensity greater than 2.93. Note that the for outdoor lighting, the highest intensity can be orders of magnitude higher than the 99.9-th percentile. Among these, Poly Haven and HDR Maps offer greater diversity across scene types. To balance the training distribution across data sources, we apply sampling weights in the ratio 3:2:2:1 in the order listed above.

For quantitative and qualitative evaluation, we use the Laval Indoor⁴ and Laval Outdoor⁵ datasets, which contain calibrated HDR panoramas of real-world indoor and outdoor scenes.

Synthetic rendering data. Similar to Object [40] and DiffusionRenderer [35], we create synthetic 3D scenes by compositing multiple 3D objects from Objaverse [10] and randomly placing them on a plane with varying plane textures. We use a filtered subset of Objaverse, containing $\sim 269,000$ 3D objects with decent geometries and material textures, to create synthetic 3D scenes. The varying plane textures are sampled from ~ 4000 PBR textures from MatSynth⁶ [54]. Each composited scene contains up to 3 sampled Objaverse objects. We additionally add up to 3 random geometry primitives (sphere, cube, and cylinder) with varying material textures to provide rich shading cues for model to learn. For each scene, we randomly render 1~4 video clips with varying camera motions (e.g., orbiting camera and oscillating camera) and environment lightings. We use a path-tracing renderer with 128 samples per pixel (spp) and the default OptiX denoiser to render the video clips with a resolution of 480×720 or 512×512 . The HDR rendering results are tone-mapped to LDR images using Blender's AgX tonemapping⁷. In total, we created $\sim 190,000$ random synthetic scenes, resulting in $\sim 260,000$ video clips with at least 16 frames per video clip.



Figure 8: Randomly sampled example images from our synthetic rendering data.

Perspective crops of HDR panorama images. We use a subset of 1251 HDR panoramas with meaningful contents from Poly Haven, HDR Maps, and HDRI Skies for the training with perspective crops. Instead of pre-processing the perspective crops from the HDR panoramas, we do the perspective crops on-the-fly during the training. The projection camera's azimuth angle is randomly sampled from $[0, 360^\circ]$ and the elevation angle is randomly sampled from -10° to 10° . The camera's field of view (FOV) is randomly sampled from 45° to 80° . The perspective crops are rendered with a resolution of 480×720 . A random tone-mapping function is applied to perspective projection crops

⁰<https://polyhaven.com/>

¹<https://hdrmaps.com/>

²<https://hdri-skies.com/>

³<https://doschdesign.com/>

⁴<http://hdrdb.com/indoor/>

⁵<http://hdrdb.com/outdoor/>

⁶<https://huggingface.co/datasets/gvecchio/MatSynth>

⁷<https://www.blender.org/>

to generate LDR images. The tone-mappings include ACES, Filmic, AgX, and Gamma-2.4 sRGB mappings. Auto-exposure (i.e., remapping the 99-th percentile intensity to 0.9) is also randomly applied to the LDR crops. For video input, we create trajectories of projection cameras by smoothly rotating the camera angle within an angular cone of 15° .

Perspective crops of LDR panorama videos. Similar to the perspective crops of HDR panorama images. We on-the-fly sample perspective crops from the LDR panorama videos. Due to the lack of HDR content, we only apply a random auto-exposure tone-mapping to the perspective crops.

B.3 Model Details and Initialization

LuxDiT is fine-tuned from the pre-trained CogVideoX-5b-I2V⁸. To adapt this model for our task, we replace the original text token with an image input token. This image token is generated in the same manner as the environment map noise token, but without adding noise. We reuse the model's existing text-processing layers (e.g., AdaLN) to process these new image input tokens. Furthermore, we extend the input projection layer to incorporate additional conditioning channels derived from the concatenated noise token; these extended channels are initialized to zero. Similarly, the output projection layer is extended to predict dual tone-mapped environment tokens, with its newly added channels initialized from the original model's weights.

B.4 User Study Details for Virtual Object Insertion

Following prior works [14, 16, 15, 58, 36], we conduct a user study on Amazon Mechanical Turk to compare our method against baseline approaches in terms of perceptual realism for virtual object insertion. Each participant is shown a pair of rendered results—one from our method and one from a baseline—and asked to assess lighting realism, focusing on shadows, reflections, and overall visual integration.

The specific instructions shown to participants are:

Instruction: Find the inserted virtual object, look at the difference, and select the more realistic image.

An AI system is trying to insert a virtual object into an image in a natural way. It aims to make the virtual object look as if it is part of the scene. There are two results: Trial A and Trial B, and the virtual object is located in the center of each image. Please zoom in to compare the differences between the two images, and pay attention to the lighting effects such as the reflections and shadows.

Which one looks more realistic?

- ☐ A
- ☐ B

Participants are required to use a monitor 24 inches or larger. Image pairs are randomly shuffled to prevent bias. Following [36], we repeat the user study three times, and recruited 11 unique participants for each experiment. We compute the percentage of *images* for which users preferred our method over the baseline, and report the average user preferences for three repeated experiments. In total, the study includes $11 \times 3 \times 11 \times 3 = 1089$ individual comparisons.

B.5 Three-sphere Evaluation Protocol

We adopt the three-sphere rendering setting described in StyleLight [55], with evaluation scripts provided by DiffusionLight⁹.

For the Laval Indoor dataset, we use the same set of HDR environment maps and corresponding perspective crops as DiffusionLight. We resize and crop the input image to 480×720 for our model. For Laval Outdoor and Poly Haven environment maps, we generate perspective crops using a fixed horizontal camera with a 60° field of view and a resolution of 480×720 . For Laval Outdoor, we apply auto-exposure by scaling the 50th percentile intensity to 0.5.

⁸<https://huggingface.co/THUDM/CogVideoX-5b-I2V>

⁹<https://github.com/DiffusionLight/DiffusionLight-evaluation>

C Additional Experiments

C.1 Array-of-Spheres Evaluation

Following prior work [61, 8], we evaluate our method using the array-of-spheres protocol, which renders a grid of diffuse spheres on a ground plane using the predicted environment map.

We use 2,240 perspective crops from 224 Laval Indoor panoramas, provided by DiffusionLight¹⁰. All input images are resized to 512×512 to match our model input. Quantitative results are shown in Table 9 and qualitative results in Fig. 9.

While our method performs slightly below specialized systems like Weber *et al.* [61] and EMLight [69], it remains competitive—despite not being trained on Laval Indoor. Notably, it outperforms StyleLight [55] and DiffusionLight [45], demonstrating strong generalization across lighting domains.

Table 9: Scores on indoor array-of-spheres.

Method	si-RMSE ↓	AE ↓
EverLight [8]	0.091	6.36
StyleLight [55]	0.123	7.09
Weber et al. [61]	0.081	<u>4.13</u>
EMLight [69]	0.099	3.99
DiffusionLight [45]	0.090	5.25
Ours	<u>0.089</u>	4.90

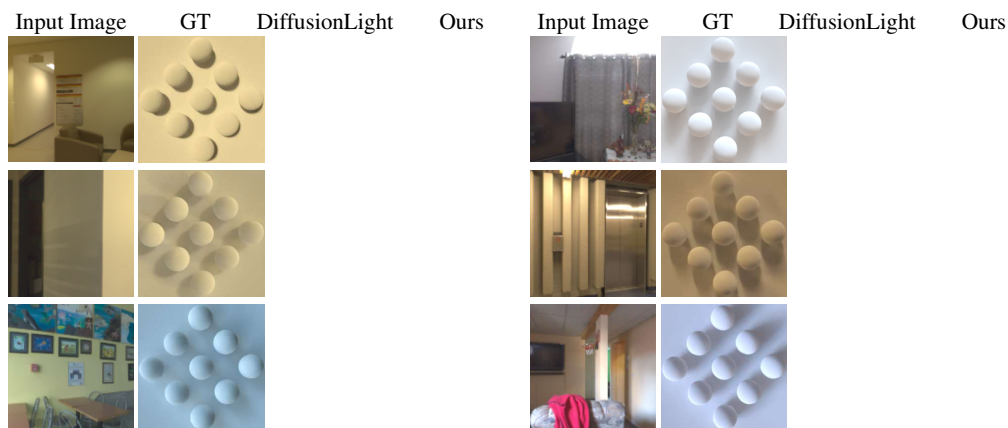


Figure 9: Visual results on array-of-spheres protocol.

C.2 Lighting Estimation with the Cube++ Dataset

We also evaluated our method on the Cube++ dataset [12], specifically designed for illumination estimation and color constancy. This dataset includes illumination information annotated by the SpyderCube¹¹. For our experiment, we selected 100 processed JPEG images from Cube++. We then applied both DiffusionLight and our method to estimate the illumination from each image. Subsequently, we rendered the left and right white faces of the SpyderCube under the estimated illumination, assuming purely Lambertian diffuse surfaces. To prevent information leakage from the SpyderCube in the input images, we masked out the SpyderCube from the tested images and inpainted the masked region using LaMa [51]. We then compared the rendered face colors to the colors sampled directly from the SpyderCube JPEG images. Table 10 presents the RMSE and angular errors, demonstrating that our method clearly outperforms DiffusionLight, achieving angular errors of less than 5° on both faces. Visual comparison results are further illustrated in Fig. 10.

Table 10: Scores on SpyderCube white face rendering on Cube++ dataset.

Method	RMSE ↓		AE ↓	
	Left	Right	Left	Right
D.Light [45]	0.044	0.035	7.221	5.741
Ours	0.024	0.025	3.985	4.003

¹⁰https://github.com/DiffusionLight/image-array_of_spheres

¹¹<https://www.datacolor.com/spyder/products/spyder-cube/>



Figure 10: Visual results on Cube++ dataset. We show the rendered two white cube faces, mirror ball, and matte silver ball from our method and DiffusionLight for visual comparison.

C.3 Lighting Estimation from Foreground Objects

Since our model is trained on object-centric synthetic rendering data, we can also apply it to estimate lighting from foreground objects. We selected 4 NeRF synthetic objects [41] and 4 real-world objects [53], aiming to estimate lighting from videos containing nine consecutive rendering views.

We qualitatively compare LuxDiT with optimization-based inverse rendering methods [42, 18] that reconstruct 3D geometry and lighting from full NeRF scenes. Using the ground truth camera poses, we rotate each frame’s estimated lighting into the global coordinate system and average across frames to produce the final environment map.

Qualitative results are shown in Fig. 11. On mostly diffuse objects like lego and hotdog, our method recovers highlight directions accurately, enabling shadow rendering consistent with the input. For glossy objects like mic and ficus, our model estimates lighting nearly identical to the ground truth. While these HDR environment maps are included in our training set, the NeRF scenes differ significantly from our synthetic renderings (see Fig. 8), indicating that our model leverages shading cues and learned priors rather than direct memorization. In contrast, optimization-based baselines struggle to capture high-frequency lighting detail and often introduce noise and artifacts in lighting.

We further tested our method on real-world foreground objects from the Objects-with-Lighting dataset [53], which provides ground truth distant environment lighting. Similar to the NeRF synthetic scene setup, the estimated lighting was then aligned into the global coordinate system using ground truth camera poses. We compared our approach to NeuS+Mitsuba [56, 26], the top-performing method on this dataset [53]. The metrics, using the three-sphere protocol, are presented in Table 11, with visual results in Fig. 12.

While our model performs well overall, minor errors remain, *e.g.* color shifts in the NeRF Lego scene (Fig. 11) and a slightly higher si-RMSE compared to NeuS+Mitsuba (Table 11). We believe combining our generative model with optimization-based methods could further enhance lighting estimation, which we leave for future work.

Table 11: Comparison of our method with NeuS+Mitsuba on Objects with Lighting datasets.

Method	Scale-invariant RMSE ↓			Angular Error ↓			Normalized RMSE ↓		
	Diffuse	Matte	Mirror	Diffuse	Matte	Mirror	Diffuse	Matte	Mirror
NeuS+Mitsuba	0.082	0.232	0.424	3.145	3.383	3.526	0.180	0.545	0.717
Ours	0.086	0.253	0.482	1.262	1.594	2.000	0.153	0.339	0.479

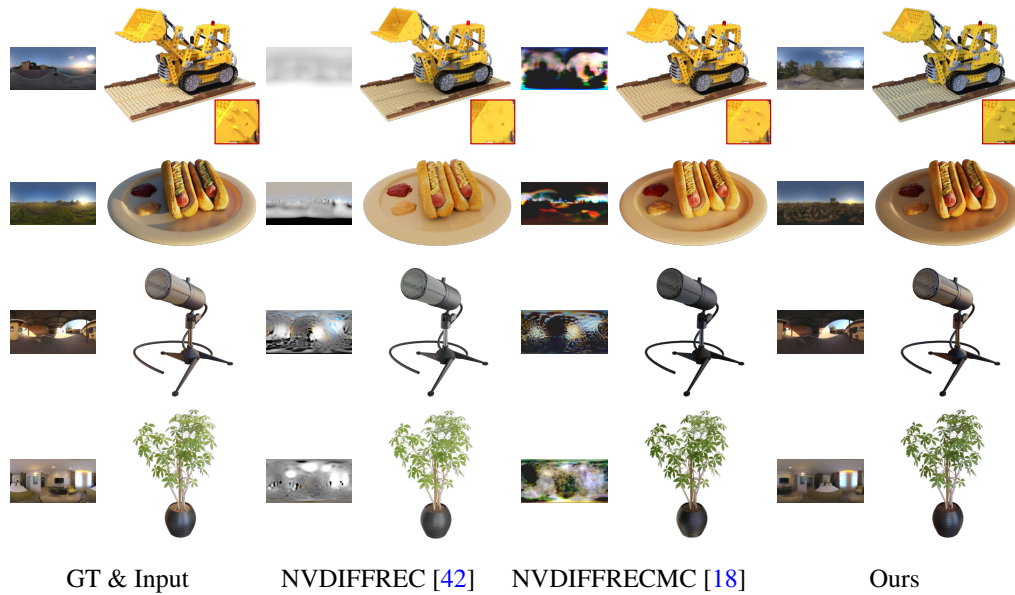


Figure 11: Lighting estimation from the NeRF synthetic objects. We use the estimated lighting from different methods to re-render the original NeRF Blender scenes.

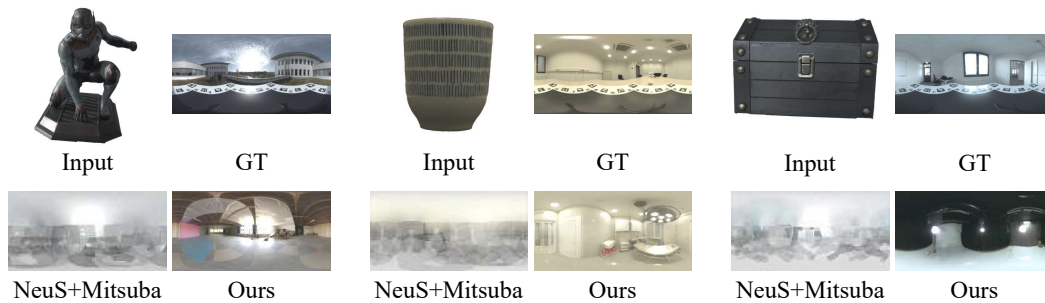


Figure 12: Lighting estimation from the masked real objects from Objects with Lighting.

C.4 Additional Ablations

C.4.1 The Choice of the HDR Fusion Model

As detailed in Sec. 4.1, a lightweight MLP ψ is employed to merge the dual-tonemapped environment maps, E_{ldr} and E_{log} , thereby reconstructing the HDR environment map $\hat{E}$. There are also alternative fusion methods, such as using a more complex CNN model to incorporate adjacent pixel information for HDR fusion, or applying a rule-based approach with explicit inverse equations. To justify our choice of a simple MLP, we evaluate various HDR fusion techniques, including MLP, CNN, and a rule-based method. The CNN model has an identical number of layers to our MLP model, using 3×3 convolution kernels across layers. The rule-based method involves applying the inverse Reinhard map for lights with intensity below 8, a linear interpolation between Reinhard and log maps for intensities ranging from 8 to 16, and exclusively the log map for intensities exceeding 16.

Table 12 presents the RMSE results on testing Polyhaven HDRIs. All three methods demonstrate comparable accuracy, with the MLP approach exhibiting a slight advantage. Compared to the rule-based approach, we believe the neural approach can better handle numerical inconsistency after image uint8 quantization, and the potential data range overflow (e.g., lights beyond the pre-defined maximum intensity 10000).

Table 12: Comparison on different HDR fusion approaches.

	MLP	CNN	Rule
RMSE ↓	11.55	11.74	11.71

C.4.2 The Impact of LoRA on Synthetic Scenes

Section 5.5 demonstrates the impact of varying LoRA scales (0.0 to 1.0) on the predicted lighting content of real-world images. This ablation study, conversely, investigates how our LoRA model, trained with real images, affects the lighting estimation of synthetic foreground objects. Table 13 presents the angular errors using a three-sphere evaluation, and Fig. 16 provides the visual results.

Table 13: Ablation study on impact of LoRA scale on synthetic foreground objects.

LoRA Scale	0.00	0.25	0.50	0.75	1.00
Diffuse ↓	1.594	1.737	2.170	3.832	3.937
Matte ↓	2.068	2.311	2.914	5.322	5.891
Mirror ↓	3.405	3.690	4.342	6.783	7.400

In contrast to the ablation performed on scene images, a larger LoRA scale leads to lower lighting estimation accuracy. As Fig. 16 illustrates, increasing the LoRA scale causes foreground content to gradually appear on the estimated environment map, which is consistent with our LoRA model's behavior. Nevertheless, the estimated highlights remain consistent across different LoRA scales.

D Additional Results

We provide additional visual results in this section to further support the claims made in the main paper.

- **Model Ablation and LoRA Scale:** Figure 13 details the ablation study on our model's design and the exploration of different LoRA scales.
- **Camera Parameter Variations:** Figures 14 and 15 show lighting estimation performance when varying camera field of view (FOV) and elevation angles, respectively.
- **Three-Sphere Rendering Evaluations:** Figures 17, 18, and 19 display further lighting estimation outcomes using the three-sphere rendering protocol on the Laval Indoor, Laval Outdoor, and Poly Haven datasets.
- **Virtual Object Insertion:** Figures 20 and 21 illustrate additional virtual object insertion results on Poly Haven panorama crops and Waymo driving scenes.

Input Image	Model Ablation	LoRA Scale Exploration			
		LoRA 0.0	LoRA 0.1	LoRA 0.2	LoRA 0.4
	channel concat.				
	w/o syn. data				

Figure 13: Model design ablation and LoRA scale exploration. The “Model Ablation” column shows the results of our two model design variants: 1) channel concatenation and 2) training without synthetic rendering data. The “LoRA Scale Exploration” columns show the visual results of our model with different LoRA scales.

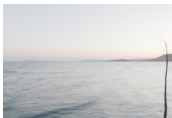
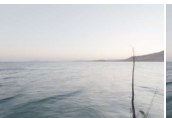
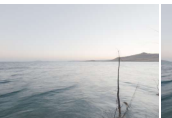
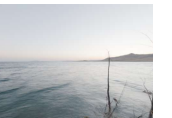
FOV	45°	50°	60°	70°	75°
GT					
					

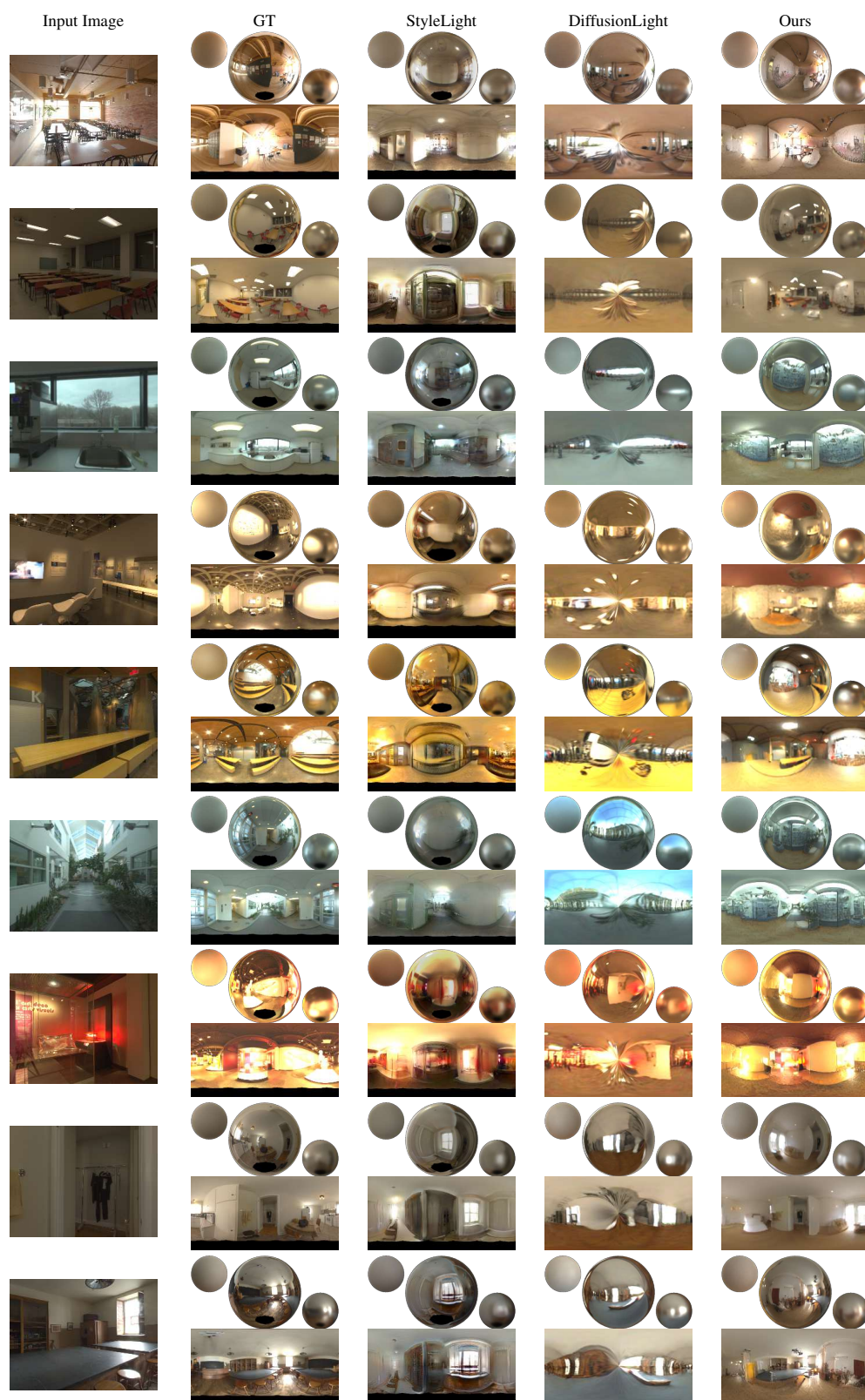
Figure 14: Lighting estimation from input images with varying camera FOV.

Elevation	30° ↑	15° ↑	0°	15° ↓	30° ↓
GT					
					

Figure 15: Lighting estimation from input images with varying camera elevation.

Input Image	LoRA Scale Exploration		
	GT	LoRA 0.00	LoRA 0.25
			
			

Figure 16: LoRA scale exploration on synthetic foreground scenes.



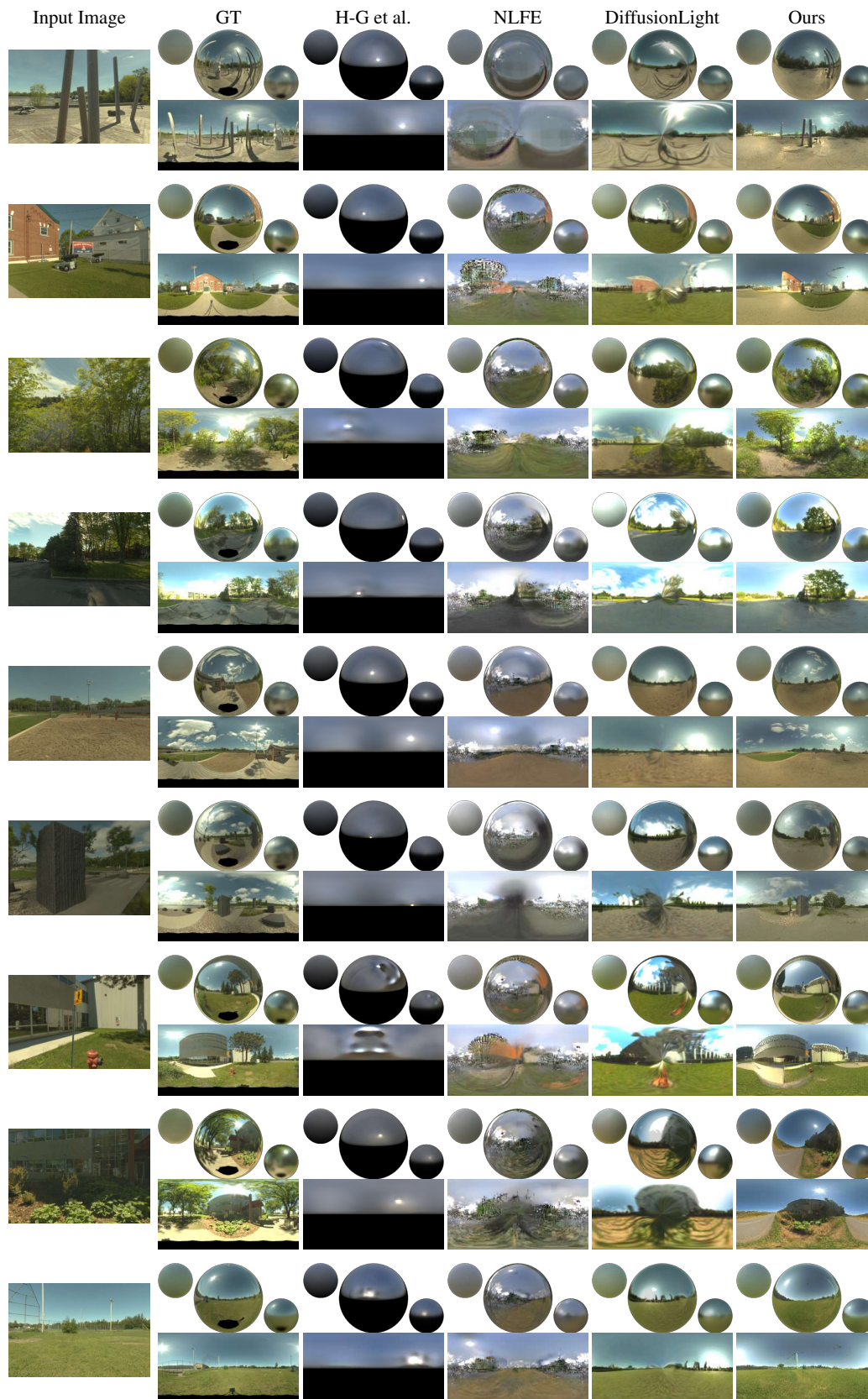


Figure 18: Additional qualitative results on Laval Outdoor dataset.

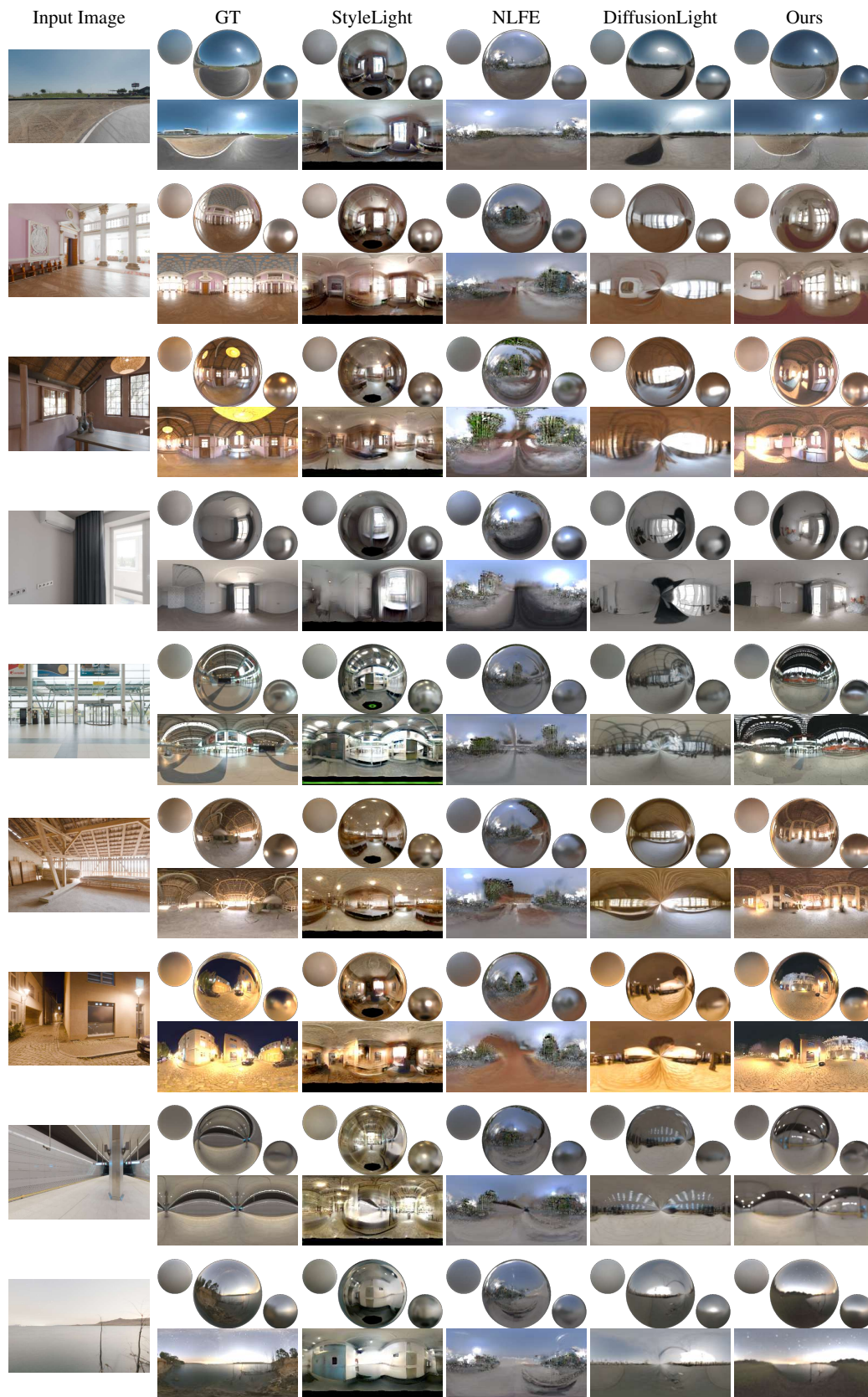


Figure 19: Additional qualitative results on Poly Haven dataset.

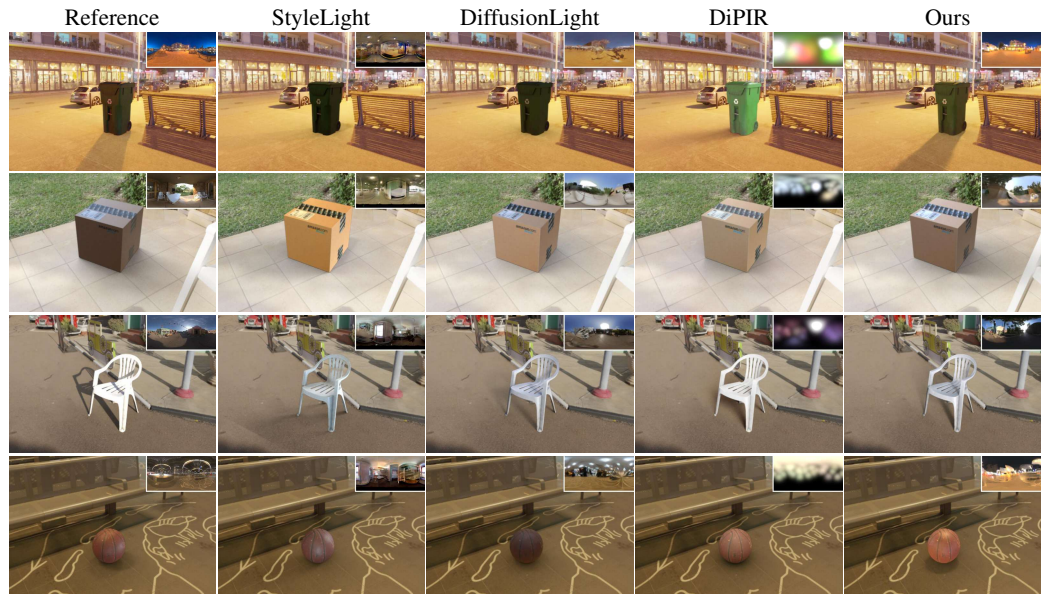


Figure 20: Additional virtual object insertion on Poly Haven perspective crops.



Figure 21: Additional virtual object insertion on Waymo driving scenes.