

Temporally Stable Generative Illumination with a One-Step Diffusion Model

Harish Anand^{*}, Alexandr Kuznetsov^{*}, Sungye Kim^{*✉}, Wojciech Uss^{},
Wojciech Kaliński^{}, and Rama Harihara^{}

Advanced Micro Devices, Inc.

Abstract. The rapid rise of generative image synthesis has intensified interest in the graphics and vision community in producing realistic, controllable illumination. We present a generative method for screen-space global illumination (GI) that produces geometrically aligned, temporally stable indirect lighting using a single-step latent diffusion model. Unlike iterative and video diffusion methods that are costly and hard to control, our model conditions a one-step image generator on scene-intrinsic signals and sparse lighting hints for the current frame, encoded via adapter modules. To address flicker without the overhead of video diffusion, we introduce a temporal VAE decoder that plugs into a latent diffusion pipeline at inference time, delivering long-range temporal coherence while preserving spatial fidelity. On test data, our approach synthesizes high-quality indirect illumination well-aligned to inputs and improves temporal stability. Our quality evaluation demonstrates that our method generates visually and quantitatively plausible illumination in generative forward rendering, outperforming state-of-the-art methods.

Keywords: Diffusion Models · Illumination Generation · Image Synthesis · Temporal Stability

1 Introduction

Physically based image synthesis plays a crucial role in computer vision and computer graphics. The rendering equation [27] models light transport in the environment, enabling the generation of photorealistic images. Ray tracing and path tracing are widely used in graphics workloads to simulate these light-matter interactions. The quality of physically based rendering depends on the complexity of the geometry and the accuracy of material models, and achieving photorealism from 3D scene descriptions often incurs a high computational cost. To reduce this cost, approximation techniques based on screen-space representations [30, 45] have been studied, offering benefits such as reduced rendering cost that is largely independent of scene complexity. Such screen-space representations can be obtained from rendering engines (e.g., G-buffers) in computer

^{*} Equal contribution.

[✉] Corresponding author: sungye.kim@amd.com

graphics or reconstructed from data in computer vision (e.g., intrinsic decomposition, inverse rendering).

Neural rendering introduces new approaches to photorealistic rendering pipelines by learning scene representations from large datasets. These methods can enhance the realism of synthetic images [44] and generate realistic illumination [20, 39, 52] by leveraging intrinsic representations of geometry and materials.

Recent advances in image and video generation with large diffusion models [23, 46, 48] demonstrate the ability to produce images with visually appealing content and convincing illumination without explicit scene descriptions. Strong priors learned from large-scale datasets have enabled this breakthrough, ushering in a new era of generative neural rendering. A few recent works have shown the use of diffusion models to generate controllable lighting effects [17, 31, 49] and specific phenomena such as caustics [53]. Other studies apply diffusion models to intrinsic image decomposition from color images or videos, and subsequently synthesize new images from intrinsic representations of material, geometry, and illumination [32, 62]. Despite these advances, precisely controlling the generated content to align with input conditions remains challenging. In particular, generating realistic illumination while respecting geometric and material constraints remains an active area of research in generative rendering and relighting. In addition, the iterative denoising process in diffusion models requires substantial time to produce an image, which limits practical applications. For videos or image sequences, temporal stability is a key determinant of perceived quality. While video diffusion models [6] can improve temporal stability, they typically incur high computational and memory costs.

We are motivated by the success of generative models [8, 41] that produce high-fidelity videos of the physical world. Accordingly, we propose generating realistic illumination using diffusion models conditioned on geometry, materials, and spatially sparse lighting. Unlike prior work [21, 32, 62], we focus on forward rendering and leverage intrinsic signals from rendering engines in synthetic scenes. To address the aforementioned challenges, we employ a single-step diffusion model that generates realistic illumination while preserving geometric alignment with the input conditions. To achieve temporal stability across image sequences without video diffusion models, we introduce a temporal variational autoencoder (TVAE) decoder that plugs into latent image diffusion pipelines, enabling temporally stable long-sequence generation. Key contributions of our work are:

- We propose an approach that generates realistic illumination on an input stream of images using a one-step image diffusion model while preserving alignment with the input conditions.
- We introduce a Temporal VAE decoder that enables the synthesis of image sequences with high temporal stability and can be integrated with latent image diffusion models.
- We demonstrate that our method generates temporally stable illumination across long video sequences and diverse test scenes, including complex geometry, multiple light sources, and camera movement.

2 Related Work

Neural Illumination. Neural networks, as high-dimensional mapping functions, have been used in computer graphics and vision to assist or replace components in producing high-quality images. Some prior works leverage neural networks to approximate illumination effects in screen-space [20, 39, 52, 57, 66], and many methods use G-buffers as inputs. G-buffers from the rendering pipeline or intrinsic images from inverse rendering provide models with essential information such as geometry (normal, depth) and material properties (albedo, roughness, metallic). Some studies [20, 52] also incorporate direct lighting as input to GAN-based networks [52] or UNet-based approaches [20] to predict indirect illumination and soft shadows. For state-of-the-art advances in neural rendering, we refer to comprehensive surveys [51].

Our proposed approach leverages generative diffusion models to produce indirect illumination. We demonstrate the method on complex scenes with diverse objects and lighting conditions.

Diffusion Models for Image Generation. In recent years, research on generative diffusion models [6, 23, 46, 48] has grown exponentially and provides new methods for high-quality image generation, achieving better sample quality and distribution coverage than GANs [15]. Diffusion models progressively refine Gaussian noise into high-quality images and offer remarkable quality advantages in both unconditional and conditional image generation tasks. Several studies use diffusion models to generate specific illumination effects such as reflections [16, 17, 56], shadows [49] and caustics [53], as well as to denoise Monte Carlo noise in path-traced images [54]. For a broader overview and applications of diffusion models, we refer to comprehensive surveys [3, 9, 60].

More recently, several studies use diffusion models as new tools for inverse rendering to estimate intrinsic properties such as geometry (normal, depth), materials (albedo, roughness, metallic), and illumination (irradiance, environment map) from a color image [11, 32, 35, 36, 62, 65]. The intrinsic images are then used to edit, composite, and relight [10, 11, 13, 18, 21, 22, 25, 33, 43, 58, 61] the color image to synthesize new images. RGB $\leftrightarrow$ X [62] uses latent-space image diffusion models to perform intrinsic decomposition (RGB $\rightarrow$ X) and forward rendering (X $\rightarrow$ RGB). Similarly, DiffusionRenderer [32] estimates intrinsic images in an inverse rendering stage and generates new color images using an environment map for lighting in a forward rendering stage, employing video diffusion models to provide temporal consistency. UniRelight [21] combines such two stages into one to relight an input video using an environment map, leveraging a DiT video model. Video diffusion models are aware of temporal correlations and motion priors from the input video; however, current video diffusion models require substantial computational power and memory [59].

Our work focuses on forward rendering to generate indirect illumination using a diffusion model. Unlike prior methods, our approach uses a latent-space diffusion model for image generation while providing strong alignment with geometric conditions and improved temporal stability across image sequences.

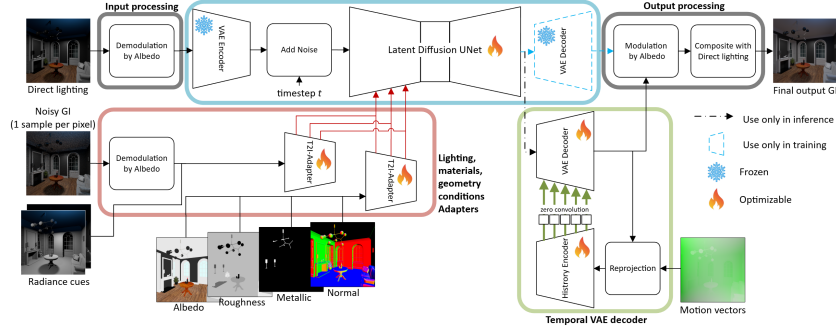


Fig. 1: Overview of our method. Given an initial lighting hint from direct lighting, our approach generates global illumination using a conditional one-step diffusion model. Multiple conditioning inputs are encoded via adapter modules and injected into the diffusion model. We use geometry (normals), material properties (albedo, roughness, metallic), and lighting cues (direct lighting, spatially sparse GI, radiance cues) as conditioning inputs. The proposed Temporal VAE decoder ensures temporal consistency over long sequences. We train the Temporal VAE decoder separately and replace the frozen VAE decoder with it at inference time. A diagram of TVAE training appears in the supplementary material.

3 Our Methodology

We formulate the screen-space approximation of global illumination as a conditional image generation task using multiple conditions of geometry, material properties, and lighting cues. As shown in Fig. 1, our approach leverages a latent diffusion model (LDM), trained on large-scale data, to exploit strong priors for high-fidelity image synthesis. Our method uses direct lighting as an initial lighting hint and generates the global illumination (GI) output. We align generation with several conditioning signals, such as sparse lighting (noisy GI), radiance cues [20, 61], materials (albedo, roughness, metallic), and geometry (normals), by encoding them with T2I-Adapter [38]. Unlike prior work [62], we address temporal stability across generated image sequences by replacing the original VAE decoder with our Temporal VAE decoder at inference time. We describe further details in the following sections.

3.1 Input and Output Processing

During input processing, we demodulate the direct lighting by dividing by the albedo, which removes material color and textures and helps the diffusion model focus on illumination generation. Since the SD-VAE is originally designed for low-dynamic range, non-linear image distributions, we transform the radiance values from a high-dynamic-range linear space to the range $[0.0, 1.0]$ using μ -Law tonemapping ($\mu=5011$), as suggested in prior work [28], before passing them to the VAE encoder. To ensure radiance values remain within $[0.0, 1.0]$ in the tone-mapped space, the compression factor is calibrated so that the maximum

luminance (1000 nits) maps to 1.0, which prevents overflow in the tone-mapped output. The VAE decoder therefore produces outputs in this non-linear (tone-mapped) space. Recent research [7, 55] has addressed direct generation of HDR images using diffusion models, but such approaches are outside the scope of our contribution. During output processing, we apply an inverse μ -Law tonemap to convert the VAE decoder outputs back to linear space, followed by albedo modulation to restore texture details and material color. Finally, we composite the result with the initial direct lighting to generate the final output.

3.2 Conditional One-Step Diffusion Model

One-Step Diffusion Model. We use a single-step diffusion model that is capable of producing high-quality images in a single denoising step. Although diffusion models offer high-fidelity image generation, their multi-step sampling is typically slow. SD-Turbo and SDXL-Turbo are distilled versions of SD [46] and SDXL [42] by adversarial diffusion distillation [47], enabling the foundational diffusion models to generate images within one or two sampling steps while maintaining fidelity. As shown in Fig. 1 (blue box), we build on Stable Diffusion 2.1-Turbo and fine-tune the diffusion UNet using our dataset while the VAE remains frozen. During UNet fine-tuning, we use the widely adopted SD-VAE [46], which we then replace with our Temporal VAE decoder during inference (Sec. 3.3). The input to the UNet is a noisy latent of direct lighting at the final timestep in the distilled one-step diffusion model. The UNet learns to generate latents of indirect illumination, which are decoded by the VAE decoder and composited with direct lighting to produce the global illumination output.

Multi-Condition Adapter. Standard image-to-image pipelines provide conditioning signals through a noisy input image that is subjected to a forward diffusion process. We find that our task benefits from multiple conditioning, both via the noisy input and via adapters that inject features at different levels of the UNet. Direct lighting, as an initial lighting hint, is encoded to the latent space using the VAE encoder and provides conditioning through the UNet input layer. We empirically find the T2I-Adapter [38] to be more effective for adapting the model to target images. Conditioning via the noisy input and multi-level feature injections into the UNet performs better, likely because directly concatenating all conditioning signals to the noisy input shifts the diffusion UNet’s input distribution and perturbs its behavior. In this work, as shown in Fig. 1 (red box), geometry (normals), material properties (albedo, roughness, metallic), and lighting cues (noisy GI, radiance cues [20, 61]) serve as inputs to the T2I-Adapters. The features from the T2I-Adapters are then equally added at different levels of the UNet for conditional generation.

3.3 Temporal VAE Decoder

Temporal stability is a critical factor in ensuring perceived video quality. Typically, a VAE decoder decodes latent vectors independently for each frame in a

video sequence. This leads to temporally unstable results, which manifest as artifacts like flicker across frames. Enforcing temporal consistency in the diffusion UNet requires substantial architectural changes and likely retraining. Hence, we introduce a temporal VAE (TVAE) decoder to support temporal consistency when generating an infinite sequence of frames. We leverage the architecture of an existing model, sd-vae-ft-mse-original [4], by extending it and fine-tuning on our data instead of training from scratch.

As shown in Fig. 1 (green box), our TVAE takes latent vectors from the diffusion UNet along with the previous output. Instead of using the previous output directly, we reproject it using motion vectors, aligning it with the current frame and simplifying optimization. The history encoder, a small CNN composed of convolution and downsampling layers, processes the previous output at multiple scales and injects the signals into the TVAE decoder via zero convolutions. In essence, the TVAE decoder produces pixel values similar to the previous frame in regions where the current latent vectors are close to the previous output. We use the pre-trained SD-VAE as the base and optimize the TVAE decoder on our dataset while keeping the VAE encoder frozen. During inference, the original SD-VAE decoder is replaced with our TVAE decoder.

3.4 Synthetic Data Generation

We generate a comprehensive synthetic dataset of interior scenes from multiple sources [5, 19], rendered using the Blender Cycles renderer [14]. From these scenes, we render 31k frames (23k for training and 8k for testing). Each scene consists of a dynamically moving camera, static and moving light sources, and selected objects in motion. Illumination is provided by 1–30 area lights with varying sizes, intensities, and colors, while materials employ principled BSDF shaders with diverse roughness and metallic parameters to capture both diffuse and glossy reflections. The dataset includes material properties (albedo, roughness, metallic), geometry (normals), and motion vectors. For lighting cues, we generate direct lighting, single-bounce noisy GI with one sample per pixel, and radiance cues [20, 61]. All images are rendered at 512×512 resolution, and saved in a high dynamic range image format. For high-quality ground truth, reference frames are rendered at 1024×1024 resolution with high sample counts (e.g., 2k–16k samples per pixel, empirically determined), followed by OptiX [40] denoising and downsampling to 512×512 .

3.5 Training

We train the conditional one-step diffusion model and the TVAE decoder separately. For training diagrams, see the supplementary material.

Fine-tuning Diffusion Model and Adapter. We jointly fine-tune the diffusion UNet and T2I-Adapter on our dataset while keeping the pre-trained SD-VAE frozen. We add Gaussian noise ($\epsilon \sim \mathcal{N}(0, 1)$) to the latent representation of

the initial lighting hint (direct lighting) at timestep $t = 499$, following a single-step img2img diffusion pipeline. Our one-step diffusion model matches the configuration used in SD-Turbo, except that we use the DDIMScheduler [48] rather than the EulerDiscreteScheduler during training and inference. Following prior work on supervised pixel-space fine-tuning for text-to-image generation [63], we optimize the final output from the VAE decoder in pixel space using the sum of L1 and LPIPS [64] losses. We use the AdamW [34] optimizer with betas of 0.9 and 0.999 and a weight decay of 0.01. The learning rate is set to 5×10^{-5} with a constant learning rate scheduler. We train our model on AMD Instinct™ MI210 GPUs for 50 epochs with a batch size of 32.

Training Temporal VAE Decoder. To train our TVAE decoder, we initialize it with parameters from the pre-trained SD-VAE decoder and freeze the SD-VAE encoder. Training uses temporal sequences of frames and we feed the latent vectors of the current frame from the frozen VAE encoder, along with the previous output reprojected to align with the current frame. For TVAE training, the ground truth of the previous frame is used as the previous output. During inference, the previous output comes from the TVAE decoder’s output of the preceding frame, and the latent vectors of the current frame come from the diffusion UNet. To facilitate efficient training via the history encoder in our TVAE, we corrupt the current latent vectors from the frozen VAE encoder by adding Gaussian noise. This enables the TVAE decoder, assisted by the history encoder, to decode the current frame’s latent vectors with temporal consistency. We optimize the TVAE using an L1 loss in pixel space. To prevent drift, we train the TVAE as a standard VAE on the DIV2K [2] dataset in parallel, with the previous frame set to zero, using L1 loss as well. We train the TVAE independently for 10 epochs on our training dataset.

4 Results

4.1 Evaluation Setup

We evaluate our technique on test data that includes scenes and images unseen during training. Since we focus on forward rendering to produce indirect illumination rather than relighting, we compare our results with two prior diffusion model-based methods, RGB $\leftrightarrow$ X [62] and DiffusionRenderer [32], using publicly available pre-trained models. For RGB $\leftrightarrow$ X, we use our G-buffer images (albedo, normals, roughness, metallic) as conditioning inputs to feed high-quality conditions and use RGB $\rightarrow$ X intrinsic decomposition to generate an irradiance image from the ground truth global illumination image. We observe that this produces better quality than using intrinsic images generated from RGB $\rightarrow$ X intrinsic decomposition. For DiffusionRenderer, we split our long image sequences into multiple 24-frame videos, as the pre-trained model requires 24-frame videos as input. We perform DiffusionRenderer’s inverse rendering for each video to generate intrinsic conditions and use DiffusionLight-Turbo [12] to generate environment maps for each frame from its ground truth global illumination, ensuring that the environment maps contain all lighting information from the frames. We then

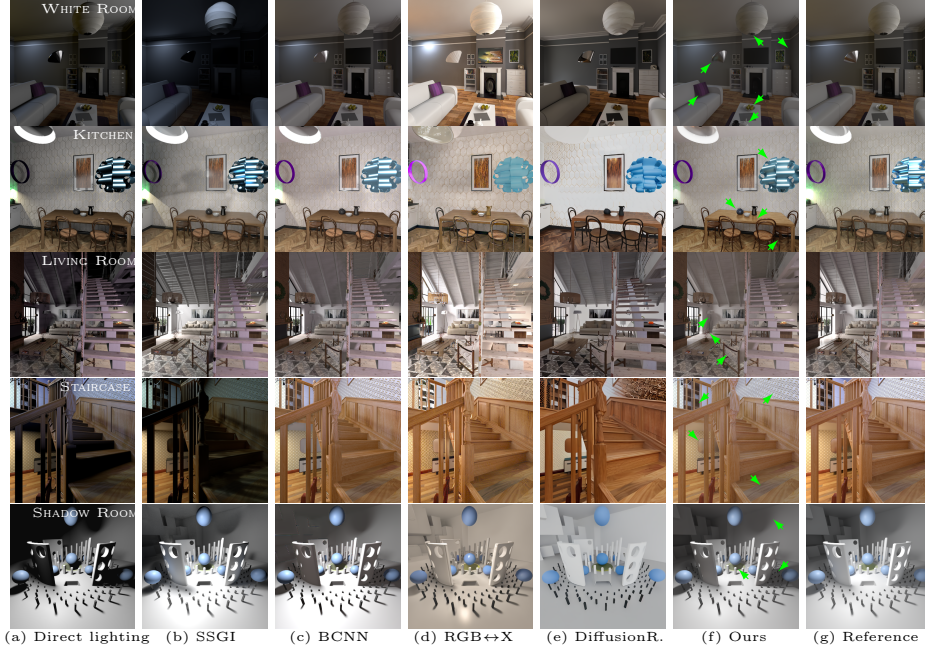


Fig. 2: Quality comparison on the test dataset. Direct lighting (a) is used as the initial lighting hint for our method. Reference (g) is path-traced global illumination rendered with the Blender Cycles renderer. We compare against SSGI (b), rendered with the Blender Eevee renderer, BCNN [57] (c), a neural GI method, and two generative methods, RGB↔X [62] (d) and DiffusionRenderer [32] (e). Arrows ($\rightarrow$) are overlaid on our results (f) to highlight areas where our method produces plausible results that are closer to the reference.

run DiffusionRenderer’s forward rendering for all videos and merge the output videos into a long sequence. Furthermore, we evaluate our results against screen-space global illumination (SSGI) from Blender Eevee renderer 4.5 LTS [14] and BCNN [57], a neural GI method that we trained on our dataset.

For quantitative evaluation, we show PSNR $\uparrow$, SSIM $\uparrow$, and LPIPS $\downarrow$. We also measure temporal stability using FovVideoVDP¹ [37], a video quality metric that captures temporal distortions and flicker perceptible to humans.

4.2 Quality Evaluation

Figure 2 compares our results with other techniques on our test dataset. Figure 2 (a) shows direct lighting between light sources and geometry, which serves as the initial lighting hint for our method. Figure 2 (b) shows screen-space global illumination from the Blender Eevee 4.5 renderer that approximates the ground truth

¹ FovVideoVDP v1.2.2, 75.4 [pix/deg], Lpeak=200, Lblack=0.5979 [cd/m²], non-foveated

Table 1: Quantitative evaluation using PSNR, SSIM, LPIPS, and FovVDP metrics on the test data in Fig. 2. The best scores are shown in bold.

	Metric	BCNN	RGB $\leftrightarrow$ X	DiffusionRenderer	Ours
WHITE R.	PSNR $\uparrow$	33.18	10.21	18.70	33.80
	SSIM $\uparrow$	0.97	0.65	0.79	0.98
	LPIPS $\downarrow$	0.04	0.26	0.13	0.04
	FovVDP $\uparrow$	6.8811	0.6669	3.0763	7.0661
KITCHEN	PSNR $\uparrow$	21.93	16.68	13.46	23.96
	SSIM $\uparrow$	0.88	0.53	0.58	0.91
	LPIPS $\downarrow$	0.11	0.36	0.38	0.10
	FovVDP $\uparrow$	6.0229	4.5451	3.8247	6.4124
LIVING R.	PSNR $\uparrow$	21.41	14.43	16.33	23.63
	SSIM $\uparrow$	0.83	0.58	0.57	0.86
	LPIPS $\downarrow$	0.15	0.28	0.27	0.14
	FovVDP $\uparrow$	5.1514	3.3022	2.9937	5.8753
STAIRCASE	PSNR $\uparrow$	25.86	18.98	13.84	30.23
	SSIM $\uparrow$	0.95	0.60	0.50	0.97
	LPIPS $\downarrow$	0.05	0.22	0.27	0.04
	FovVDP $\uparrow$	6.1955	4.1449	3.1187	6.4422
SHADOW R.	PSNR $\uparrow$	19.39	14.34	12.11	21.12
	SSIM $\uparrow$	0.90	0.79	0.74	0.95
	LPIPS $\downarrow$	0.10	0.16	0.23	0.05
	FovVDP $\uparrow$	4.8756	2.6496	2.2668	5.8838

global illumination. Figure 2 (g) is the reference global illumination rendered from the Blender Cycles renderer. Note that the three scenes shown in Fig. 2 (WHITE ROOM, KITCHEN, LIVING ROOM) are unseen during training. The other scenes (STAIRCASE, SHADOW ROOM) are partially used for training, but test data is rendered from different camera trajectories and with different objects.

Across the test data presented in Fig. 2, our results exhibit visually plausible illumination that more closely matches the reference global illumination. Our results well present the metallic object with specular highlights in WHITE ROOM, glossy reflections on the table in KITCHEN, and soft shadows in SHADOW ROOM. Moreover, our method better preserves geometric structure and details from the input conditions than the comparative baselines using diffusion models. In RGB $\leftrightarrow$ X [62], we observe noticeable hallucinations that are misaligned with the input conditions—for example, the metallic object, wall-mounted TV, and fireplace in WHITE ROOM; missing reflections on the glossy table in KITCHEN. These lead to temporal inconsistencies and flicker. Overall, RGB $\leftrightarrow$ X results are much brighter than the ground truth global illumination, even though the irradiance condition is generated from the ground truth image. DiffusionRenderer also shows hallucinations, such as distorted apples on the coffee table in WHITE ROOM and wall patterns in STAIRCASE. Lighting condition for DiffusionRenderer comes from environment maps generated from ground truth global illumination images, and the final illumination quality depends heavily on the quality of the environment maps, which we produce using DiffusionLight-Turbo [12]. Our results also outperform traditional SSGI and BCNN [57] for diffuse and glossy reflections, as shown in STAIRCASE, particularly on the ceiling and wall-mounted object in the back. See supplementary material for error map visualizations.



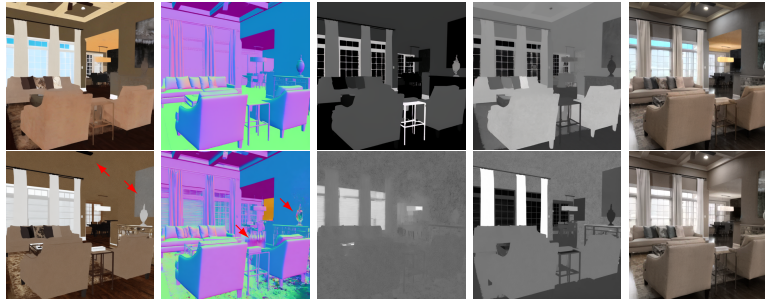
Fig. 3: Quality evaluation for real-world images compared to RGB↔X results. Our model (Ours-C) is trained using compact conditioning inputs (albedo, roughness, metallic, normals, and luminance) that can be obtained using inverse rendering of real-world data. Original images courtesy of Pexels.²

Table 1 presents quality metrics for the test data shown in Fig. 2. Our method achieves substantially better quantitative scores in PSNR $\uparrow$, SSIM $\uparrow$, and LPIPS $\downarrow$ by a large margin. It demonstrates that our results closely adhere to the input conditions, preserving geometry, color, and texture details while generating illumination. By contrast, the baselines using diffusion models exhibit noticeable hallucinations, leading to much lower metric values. Our results also achieve better metrics than BCNN by leveraging the diffusion model’s generative capability while well-aligning to conditions. For example, BCNN fails to accurately predict diffuse and glossy reflections in KITCHEN, LIVING ROOM, and STAIRCASE, resulting in low PSNR scores. We also show FovVDP $\uparrow$ for video quality in Tab. 1, computed over long sequences (e.g., 300–1000 frames) for each test data. RGB↔X generates each image without enforcing temporal consistency, resulting in lower scores across all test data. For DiffusionRenderer, which outputs 24-frame videos, we first concatenate the 24-frame output videos into a full-length video before computing FovVDP. While each 24-frame video is temporally consistent, discontinuities occur every 24 frames, which lowers the FovVDP values. BCNN is trained using a temporal loss. Our method achieves better video quality metric scores than the baselines for all test data. Please see the supplemental video for additional qualitative evidence.

4.3 Evaluation using Real-world data

Although our method is trained only on synthetic data rendered from 3D scenes, we also evaluate our model on real-world data. To this end, we train our diffusion

² <https://www.pexels.com>



(a) Albedo (b) Normal (c) Metallic (d) Roughness (e) Ours-C
Fig. 4: High-quality (top) and low-quality (bottom) intrinsics (a-d) of REAL5 in Fig. 3. Our results using these intrinsics are in (e).



Fig. 5: Evaluation without and with TVAE using OF.

UNet and condition Adapters with a compact set of conditioning inputs, such as albedo, roughness, metallic, and normal, in our training dataset, called Ours-C. Because there is no direct lighting image for real-world data, we generate a single-channel luminance image from the reference ground-truth image in our synthetic dataset and use the luminance image as a lighting hint during training. At inference time, we generate a luminance image from original real-world data to use as a lighting hint, assuming that we obtain such lighting hint from intrinsic decomposition. We use intrinsic images (albedo, roughness, metallic, and normal) generated with the inverse renderer of DiffusionRenderer [32] as geometry and materials conditioning inputs to RGB $\leftrightarrow$ X and our method (Ours-C).

Figure 3 shows results from our method alongside a comparative baseline and the original real-world images. Although our model has not been fine-tuned on any real-world dataset, it adapts to these unseen real-world images and shows good alignment with the features and illumination of the original images. In this evaluation, we observe that our results present some hallucinations in color and texture patterns, indicating the importance of the demodulation by albedo in our method (see the ablation study in Sec. 5). Such hallucinations are plausible for a real-world setting and still provide good perceptual quality. Note that our method (Ours-C) uses a single-channel luminance image as a lighting condition, but this can readily be extended to colored (3-channel) irradiance.

Robustness to imperfect intrinsics. Figure 4 shows two sets of intrinsics, high quality and low quality, generated via inverse rendering using DiffusionRenderer [32] and RGB $\leftrightarrow$ X [62], respectively. Note the missing ceiling color and painting in Fig. 4 (a), noisy normals in Fig. 4 (b), and low-quality metallic

Table 2: Quantitative evaluation on the real-world data in Fig. 3. (PSNR $\uparrow$ /SSIM $\uparrow$ /LPIPS $\downarrow$ /FovVDP $\uparrow$ (FovVDP with optical flow)).

	w/ high-quality intrinsics				w/ low-quality intrinsics			
REAL1	25.64	/	0.86	/	0.16	/	7.6627 (6.8596)	25.85 / 0.85 / 0.17 / 7.7561
REAL2	21.61	/	0.83	/	0.22	/	7.5914 (6.7521)	22.18 / 0.83 / 0.22 / 7.6991
REAL3	23.62	/	0.85	/	0.21	/	8.1052 (6.6716)	22.88 / 0.85 / 0.21 / 8.1433
REAL4	23.15	/	0.85	/	0.23	/	8.0286 (6.2275)	23.47 / 0.85 / 0.26 / 8.0453
REAL5	23.04	/	0.86	/	0.17	/	7.7276 (6.9514)	24.29 / 0.87 / 0.16 / 7.8561
REAL6	20.11	/	0.83	/	0.19	/	7.5901 (7.1617)	20.86 / 0.84 / 0.19 / 7.5711

Table 3: Runtime and GPU memory usage (PyTorch models).

512 $\times$ 512 resolution	Ours	Ours-C	RGB $\leftrightarrow$ X	DiffusionRenderer
Runtime (seconds)	0.29	0.284	3.25	52.09
GPU memory (GB)	8.68	7.97	3.76	21.81

in Fig. 4 (c). Despite these low-quality conditions, our result in Fig. 4 (e) still exhibits robust quality. Table 2 confirms this quantitatively: metrics for the real-world data in Fig. 3 are consistent across both sets with only a marginal delta, evidencing the robustness of our method.

Robustness to imperfect motion vectors. Table 2 also reports FovVDP in parentheses for our results using TVAE with optical flow (OF), generated using RAFT [50] (pretrained model: *raft-things.pth*). FovVDP values slightly decrease with OF-based TVAE, as OF is not accurate enough for reprojecting previous outputs. Nevertheless, visual quality remains plausible, with reduced high-frequency flickering when playing videos. Figure 5 shows results without and with TVAE using OF, including a new outdoor result to show how our method works. Imperfect motion vectors may cause over-smoothing, as marked with red arrows in Fig. 5, since TVAE is trained with accurate motion vectors. Leveraging OF during training TVAE would improve robustness, and making TVAE motion vector-free is also an interesting direction of future work.

4.4 Runtime Efficiency

Without incorporating inference optimizations in our implementation, our approach takes 0.29 seconds per image. Table 3 shows runtime and GPU memory for our methods and prior works on RTXTM 3090 GPU. Our key contribution is the ability to generate temporally stable illumination from a single-step diffusion model, eliminating the need for multiple denoising steps or costly video diffusion models. This provides greater runtime efficiency, being 11-179 $\times$ faster than baseline methods that require multiple denoising steps or video diffusion models.

5 Ablation Studies

Albedo demodulation: Demodulation by albedo in Fig. 1 (grey box) removes albedo (color, textures) from the input to the diffusion UNet, allowing it to focus

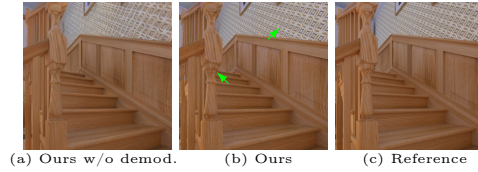


Fig. 6: Ablation of demodulation by albedo. Demodulation (in input processing) and modulation (in output processing) by albedo helps preserve texture details.

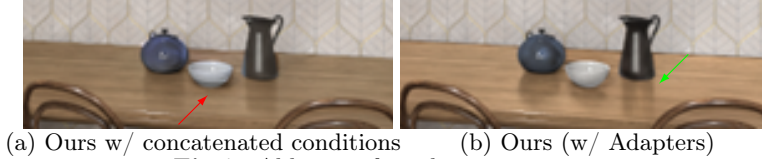


Fig. 7: Ablation of conditioning strategy.

on generating indirect illumination. Albedo modulation then reintroduces albedo into the output, preserving texture detail and color. Figure 6 (a) shows texture hallucinations when demodulation is omitted. Hence, it helps our method align with the conditioning inputs as shown in Fig. 6 (b).

Conditioning strategy: When using multiple images to condition diffusion model-based image generation, concatenation-based conditioning has showed great success in prior work [29, 32, 62]. As more conditioning images are used, we find that encoding so many images with VAE encoders and concatenating them dilutes the lighting condition signal in our approach with a single-step diffusion model. Note that the prior works use multi-step denoising diffusion models. Hence, Fig. 7 shows that our design choice with Adapters yields better reflection quality than concatenated latents of conditions. Another insight is that Adapters provide efficient feature injection into the diffusion UNet at multiple scales, enhancing controlled image generation.

Lighting hints: We use direct lighting, spatially sparse noisy GI, and radiance cues [20, 61] as lighting hints. Direct lighting is used as the initial hint through the VAE encoder, whereas noisy GI is encoded through the adapter. Although direct lighting and noisy GI are largely interchangeable with minimal overall quality differences, we observe noticeably better glossy reflections when encoding noisy GI through the adapter rather than the VAE encoder. As shown in Fig. 8 (b), our model produces stronger reflection on the glossy surfaces when direct lighting is used as the initial hint via the VAE encoder and noisy GI is supplied as an additional hint through the T2I-Adapter. This indicates that the lighting information in noisy GI is more effectively conveyed to multiple levels of the diffusion UNet when routed through the adapter.

Direct lighting as the initial hint also plays a key role in producing high-quality images. In Fig. 9, we compare our model with and without direct lighting. To this end, we use a reduced set of conditioning inputs by omitting direct lighting but use noisy GI as the initial hint. As shown in Fig. 9 (a), without direct lighting, the result exhibits a loss of detail in both geometry and illumination.

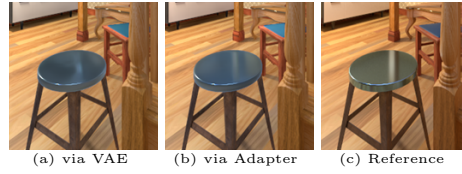


Fig. 8: Ablation of noisy GI encoding: (a) via the VAE encoder and (b) via the Adapter.

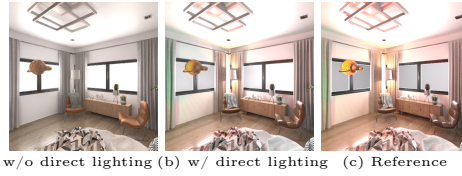


Fig. 9: Ablation (a) without and (b) with direct lighting.

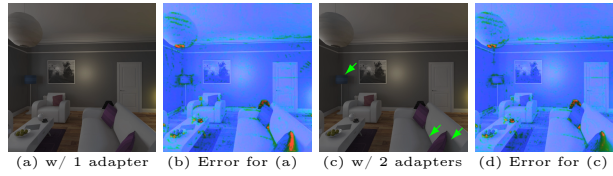


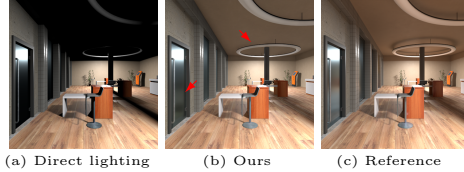
Fig. 10: Ablation of using two adapters, which produces better indirect lighting. The error maps are produced by CGVQM [26].

Dual Adapters: For the multiple conditions shown in Fig. 1 (red box), we ablate two configurations: a single T2I-Adapter for all conditions, and two T2I-Adapters—one for lighting cues and the other for material/geometry conditions. Figure 10 shows that using two adapters, with a dedicated adapter for lighting cues, produces better indirect lighting closer to the reference.

Temporal VAE: In Tab. 4, we present the video quality metric CGVQM [26], which is calibrated for different type of distortions in synthetic images and accounts for both spatial and temporal artifacts. We compute quality scores on long sequences from our test dataset, comparing video quality with and without TVAE and albedo demodulation. Overall, our method achieves comparably high quality scores. Removing TVAE, albedo demodulation, or both reduces the quality score, indicating a larger deviation from the reference. We observe a larger quality score drop without albedo demodulation, as it fails to preserve color and texture details. Note that SHADOW ROOM attains a similar score without albedo demodulation, since it does not include textures. Although the difference in quality scores between ours and without TVAE is marginal, our method with TVAE exhibits improved visual temporal stability across all test data compared with the standard SD-VAE.

Table 4: Ablation with and without TVAE and albedo demodulation on the test data in Fig. 2. Without TVAE, we run our diffusion UNet with the standard SD-VAE.

CGVQM $\uparrow$	Ours	w/o TVAE	w/o albedo demodulation	w/o both
WHITE ROOM	80.36	80.60	75.18	75.99
KITCHEN	75.97	74.74	73.86	72.86
LIVING ROOM	75.08	73.95	74.02	73.39
STAIRCASE	69.80	70.18	60.94	60.27
SHADOW ROOM	87.23	86.55	87.66	87.91

**Fig. 11:** Failure cases are marked with red arrows ($\rightarrow$).

6 Limitations

As shown in Fig. 11, indirect shadows in directly shadowed regions and reflections from distant objects remain challenging. Our results show missing color in reflections from distant objects. This partly stems from relatively small synthetic dataset with limited glossy reflections.

Our TVAE relies on the previous output reprojected using motion vectors, which are not available in real-world settings. Although TVAE is robust with optical flows as presented in Sec. 4.3, making TVAE motion vector-free would improve applicability across synthetic and real-world data.

7 Conclusion and Future work

We presented a single-step diffusion approach for generating global illumination that remains temporally stable over long sequences and aligned with intrinsic scene signals. Our key design choices—injecting lighting hints via the VAE and T2I-Adapters, and introducing a Temporal VAE decoder for stability—enable strong alignment to the conditioning inputs without sacrificing generative capacity for illumination. Experiments across diverse synthetic scenes show that the method produces plausible indirect lighting and improves temporal stability versus baselines. Ablations confirm the complementary roles of (i) adapter-conditioned noisy GI for conveying multi-bounce lighting (notably improving glossy reflections) and (ii) direct lighting as an initial hint for generating details and shadows. A compact variant trained without direct lighting shows plausible quality, indicating great general applicability to real-world images. A promising direction for future work is to explore an adaptive balance between reconstruction and generation under constrained conditioning environments, thereby enabling controllable generative flexibility based on the conservativeness of the conditioning.

References

1. Áfra, A.T.: Intel[®] Open Image Denoise. <https://www.openimagedenoise.org> (2024), Accessed June 26, 2026
2. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (July 2017)
3. Ahsan, M.M., Raman, S., Liu, Y., Siddique, Z.: A comprehensive survey on diffusion models and their applications. *Applied Soft Computing* **181**, 113470 (2025). <https://doi.org/https://doi.org/10.1016/j.asoc.2025.113470>, <https://www.sciencedirect.com/science/article/pii/S1568494625007811>, Accessed June 26, 2026
4. AI, S.: sd-vae-ft-mse-original, <https://huggingface.co/stabilityai/sd-vae-ft-mse-original>, Accessed June 26, 2026
5. Bitterli, B.: Rendering resources (2016), <https://benedikt-bitterli.me/resources/>
6. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., Jampani, V., Rombach, R.: Stable video diffusion: Scaling latent video diffusion models to large datasets (2023), <https://arxiv.org/abs/2311.15127>, Accessed June 26, 2026
7. Bolduc, C., Hold-Geoffroy, Y., Shu, Z., Lalonde, J.F.: Gaslight: Gaussian splats for spatially-varying lighting in hdr. 2025 IEEE/CVF International Conference on Computer Vision (ICCV) (2025)
8. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators (2024)
9. Cao, H., Tan, C., Gao, Z., Xu, Y., Chen, G., Heng, P.A., Li, S.Z.: A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering* **36**(7), 2814–2830 (2024). <https://doi.org/10.1109/TKDE.2024.3361474>, Accessed June 26, 2026
10. Chaturvedi, S., Ren, M., Hold-Geoffroy, Y., Liu, J., Dorsey, J., Shu, Z.: Synth-Light: Portrait relighting with diffusion model by learning to re-render synthetic faces. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2025)
11. Chen, Z., Xu, T., Ge, W., Wu, L., Yan, D., He, J., Wang, L., Zeng, L., Zhang, S., Chen, Y.: Uni-Renderer: Unifying rendering and inverse rendering via dual stream diffusion. *arXiv: 2412.15050* (2025), <https://arxiv.org/abs/2412.15050>, Accessed June 26, 2026
12. Chinchuthakun, W., Phongthawee, P., Raj, A., Jampani, V., Khungurn, P., Suwajanakorn, S.: DiffusionLight-Turbo: Accelerated light probes for free via single-pass chrome ball inpainting. In: *ArXiv* (2025)
13. Choi, J.M., Wang, A., Peers, P., Bhattad, A., Sengupta, R.: ScribbleLight: Single image indoor relighting with scribbles. *arXiv preprint arXiv:2411.17696* (2024)
14. Community, B.O.: Blender - a 3d modelling and rendering package (2018)
15. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) *Advances in Neural Information Processing Systems*. vol. 34, pp. 8780–8794. Curran Associates, Inc. (2021), https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf, Accessed June 26, 2026
16. Dhiman, A., Shah, M., Babu, R.V.: MirrorVerse: Pushing diffusion models to realistically reflect the world (2025), <https://arxiv.org/abs/2504.15397>, Accessed June 26, 2026

17. Dhiman*, A., Shah*, M., Parihar, R., Bhalgat, Y., Boregowda, L.R., Babu, R.V.: Reflecting Reality: Enabling diffusion models to produce faithful mirror reflections. In: 3DV (2025)
18. Fortier-Chouinard, F., Zhang, Z., Messier, L.E., Garon, M., Bhattad, A., Lalonde, J.F.: SpotLight: Shadow-guided object relighting via diffusion (2025), <https://arxiv.org/abs/2411.18665>, Accessed June 26, 2026
19. Games, E.: Unreal engine fab, <https://www.fab.com/>, Accessed June 26, 2026
20. Gao, D., Mu, H., Xu, K.: Neural global illumination: Interactive indirect illumination prediction under dynamic area lights. *IEEE Transactions on Visualization and Computer Graphics* **29**(12), 5325–5341 (2023). <https://doi.org/10.1109/TVCG.2022.3209963>
21. He, K., Liang, R., Munkberg, J., Hasselgren, J., Vijaykumar, N., Keller, A., Fidler, S., Gilitschenski, I., Gojcic, Z., Wang, Z.: UniRelight: Learning joint decomposition and synthesis for video relighting. *arXiv preprint arXiv:2506.15673* (2025)
22. He, M., Clausen, P., Tasel, A.L., Ma, L., Pilarski, O., Xian, W., Rikker, L., Yu, X., Burgert, R., Yu, N., Debevec, P.: DiffRelight: Diffusion-based facial performance relighting. In: *SIGGRAPH Asia 2024 Conference Papers*. SA '24, Association for Computing Machinery, New York, NY, USA (2024). <https://doi.org/10.1145/3680528.3687644>
23. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
24. Huo, Y., eui Yoon, S.: A survey on deep learning-based monte carlo denoising. *Computational Visual Media* **7**(2), 169–185 (2021). <https://doi.org/10.1007/s41095-021-0209-9>
25. Jin, H., Li, Y., Luan, F., Xiangli, Y., Bi, S., Zhang, K., Xu, Z., Sun, J., Snively, N.: Neural Gaffer: Relighting any object via diffusion. In: *Advances in Neural Information Processing Systems* (2024)
26. Jindal, A., Sadaka, N., Thomas, M.M., Sochenov, A., Kaplanyan, A.: CGVQM+D: Computer Graphics Video Quality Metric and Dataset. *Computer Graphics Forum* (2025). <https://doi.org/10.1111/cgf.70221>, Accessed June 26, 2026
27. Kajiya, J.T.: The rendering equation. *SIGGRAPH Comput. Graph.* **20**(4), 143–150 (aug 1986). <https://doi.org/10.1145/15886.15902>, <https://doi.org/10.1145/15886.15902>, Accessed June 26, 2026
28. Kalantari, N.K., Ramamoorthi, R.: Deep high dynamic range imaging of dynamic scenes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017)* **36**(4) (2017)
29. Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R.C., Schindler, K.: Repurposing diffusion-based image generators for monocular depth estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024)
30. Klehm, O., Ritschel, T., Eisemann, E., Seidel, H.P.: Bent Normals and Cones in Screen-space. In: Eisert, P., Hornegger, J., Polthier, K. (eds.) *Vision, Modeling, and Visualization* (2011). The Eurographics Association (2011). <https://doi.org/10.2312/PE/VMV/VMV11/177-182>, Accessed June 26, 2026
31. Kocsis, P., Philip, J., Sunkavalli, K., Nießner, M., Hold-Geoffroy, Y.: LightIt: Illumination modeling and control for diffusion models. In: *CVPR* (2024)
32. Liang, R., Gojcic, Z., Ling, H., Munkberg, J., Hasselgren, J., Lin, Z.H., Gao, J., Keller, A., Vijaykumar, N., Fidler, S., Wang, Z.: DiffusionRenderer: Neural inverse and forward rendering with video diffusion models. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2025)

33. Liu, Y., Luo, C., Tang, Z., Li, Y., Yang, Y., Ning, Y., Fan, L., Peng, J., Zhang, Z.: TC-light: Temporally coherent generative rendering for realistic world transfer. In: The Thirty-Ninth Annual Conference on Neural Information Processing Systems (2025)
34. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2017), <https://api.semanticscholar.org/CorpusID:53592270>
35. Luo, J., Ceylan, D., Yoon, J.S., Zhao, N., Philip, J., Frühstück, A., Li, W., Richardt, C., Wang, T.: IntrinsicDiffusion: Joint intrinsic layers from latent diffusion models. In: ACM SIGGRAPH 2024 Conference Papers. SIGGRAPH '24, Association for Computing Machinery, New York, NY, USA (2024). <https://doi.org/10.1145/3641519.3657472>
36. Lyu, L., Deschaintre, V., Hold-Geoffroy, Y., Hašan, M., Yoon, J.S., Leimkühler, T., Theobalt, C., Georgiev, I.: IntrinsicEdit: Precise generative image manipulation in intrinsic space. *ACM Transactions on Graphics* **44**(4) (2025)
37. Mantiuk, R.K., Denes, G., Chapiro, A., Kaplanyan, A., Rufo, G., Bachy, R., Lian, T., Patney, A.: FovVideoVDP: a visible difference predictor for wide field-of-view video. *ACM Trans. Graph.* **40**(4) (Jul 2021). <https://doi.org/10.1145/3450626.3459831>, Accessed June 26, 2026
38. Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., Shan, Y., Qie, X.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453* (2023)
39. Nalbach, O., Arabadzhyska, E., Mehta, D., Seidel, H.P., Ritschel, T.: Deep Shading: Convolutional Neural Networks for Screen Space Shading. *Computer Graphics Forum* (2017). <https://doi.org/10.1111/cgf.13225>, Accessed June 26, 2026
40. NVidia: Optix ai-accelerated denoiser. <https://developer.nvidia.com/optix-denoiser> (2021), Accessed June 26, 2026
41. OpenAI: Sora 2. <https://openai.com/index/sora-2/> (2025), Accessed June 26, 2026
42. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Muller, J., Penna, J., Rombach, R.: SDXL: Improving latent diffusion models for high-resolution image synthesis. *ArXiv abs/2307.01952* (2023)
43. Poirier-Ginter, Y., Gauthier, A., Philip, J., Lalonde, J.F., Drettakis, G.: Diffusion approach to radiance field relighting using multi-illumination synthesis. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)* **43**(4) (July 2024), <http://www-sop.inria.fr/reves/Basilic/2024/PGPLD24>, Accessed June 26, 2026
44. Richter, S., Haija, H., Koltun, V.: Enhancing photorealism enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PP**, 1–1 (04 2022). <https://doi.org/10.1109/TPAMI.2022.3166687>, Accessed June 26, 2026
45. Ritschel, T., Grosch, T., Seidel, H.P.: Approximating dynamic global illumination in image space. In: *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*. p. 75–82. I3D '09, Association for Computing Machinery, New York, NY, USA (2009). <https://doi.org/10.1145/1507149.1507161>, <https://doi.org/10.1145/1507149.1507161>, Accessed June 26, 2026
46. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2021)
47. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation. In: *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXXVI*. p. 87–103. Springer-Verlag, Berlin, Heidelberg (2024). https://doi.org/10.1007/978-3-031-73016-0_6, https://doi.org/10.1007/978-3-031-73016-0_6, Accessed June 26, 2026

48. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2021), <https://openreview.net/forum?id=St1giarCHLP>, Accessed June 26, 2026
49. Tasar, O., Chadebec, C., Aubin, B.: Controllable shadow generation with single-step diffusion models from synthetic data (2024)
50. Teed, Z., Deng, J.: RAFT: Recurrent all-pairs field transforms for optical flow. In: Computer Vision – ECCV 2020. vol. abs/2003.12039, pp. 402–419 (2020), <https://api.semanticscholar.org/CorpusID:214667893>, Accessed June 26, 2026
51. Tewari, A., Thies, J., Mildenhall, B., Srinivasan, P., Tretschk, E., Yifan, W., Lassner, C., Sitzmann, V., Martin-Brualla, R., Lombardi, S., Simon, T., Theobalt, C., Nießner, M., Barron, J.T., Wetzstein, G., Zollhöfer, M., Golyanik, V.: Advances in neural rendering. *Computer Graphics Forum* **41**(2), 703–735 (2022). <https://doi.org/https://doi.org/10.1111/cgf.14507>, <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14507>, Accessed June 26, 2026
52. Thomas, M.M., Forbes, A.G.: Deep Illumination: Approximating dynamic global illumination with generative adversarial network (2018), <https://arxiv.org/abs/1710.09834>, Accessed June 26, 2026
53. Uss, W., Kaliński, W., Kuznetsov, A., Anand, H., Kim, S.: Cardioid Caustics Generation with Conditional Diffusion Models. In: Ceylan, D., Li, T.M. (eds.) *Eurographics 2025 - Short Papers*. The Eurographics Association (2025). <https://doi.org/10.2312/egs.20251030>, Accessed June 26, 2026
54. Vavilala, V., Vasanth, R., Forsyth, D.: Denoising monte carlo renders with diffusion models. In: *ACM SIGGRAPH 2024 Posters*. SIGGRAPH '24, Association for Computing Machinery, New York, NY, USA (2024). <https://doi.org/10.1145/3641234.3671026>, Accessed June 26, 2026
55. Wang, C., Xia, Z., Leimkuhler, T., Myszkowski, K., Zhang, X.: Lediff: Latent exposure diffusion for hdr generation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 453–464 (June 2025)
56. Winter, D., Cohen, M., Fruchter, S., Pritch, Y., Rav-Acha, A., Hoshen, Y.: Object-Drop: Bootstrapping counterfactuals for photorealistic object removal and insertion. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) *Computer Vision – ECCV 2024*. pp. 112–129. Springer Nature Switzerland, Cham (2024)
57. Xin, H., Zheng, S., Xu, K., Yan, L.: Lightweight bilateral convolutional neural networks for interactive single-bounce diffuse indirect illumination. *IEEE Transactions on Visualization & Computer Graphics* **28**(04), 1824–1834 (apr 2020)
58. Xing, X., Groh, K., Karaoglu, S., Gevers, T., Bhattad, A.: Luminet: Latent intrinsics meets diffusion models for indoor scene relighting. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 442–452 (2025)
59. Xing, Z., Feng, Q., Chen, H., Dai, Q., Hu, H., Xu, H., Wu, Z., Jiang, Y.G.: A survey on video diffusion models. *ACM Comput. Surv.* **57**(2) (Nov 2024). <https://doi.org/10.1145/3696415>, <https://doi.org/10.1145/3696415>, Accessed June 26, 2026
60. Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., Yang, M.H.: Diffusion models: A comprehensive survey of methods and applications. *ACM Comput. Surv.* **56**(4) (Nov 2023). <https://doi.org/10.1145/3626235>, <https://doi.org/10.1145/3626235>, Accessed June 26, 2026
61. Zeng, C., Dong, Y., Peers, P., Kong, Y., Wu, H., Tong, X.: DiLightNet: Fine-grained lighting control for diffusion-based image generation. In: *ACM SIGGRAPH 2024 Conference Papers* (2024)

62. Zeng, Z., Deschaintre, V., Georgiev, I., Hold-Geoffroy, Y., Hu, Y., Luan, F., Yan, L.Q., Hašan, M.: RGB $\leftrightarrow$ X: Image decomposition and synthesis using material- and lighting-aware diffusion models. In: ACM SIGGRAPH 2024 Conference Papers. SIGGRAPH '24, Association for Computing Machinery, New York, NY, USA (2024). <https://doi.org/10.1145/3641519.3657445>, Accessed June 26, 2026
63. Zhang, C., Motwani, S., Yu, M., Hou, J., Juefei-Xu, F., Tsai, S., Vajda, P., He, Z., Wang, J.: Text-to-image generation post-training with pixel-space loss. In: Proceedings of the 3rd International Workshop on Rich Media With Generative AI. p. 30–38. RichMediaGAI '25, Association for Computing Machinery, New York, NY, USA (2025). <https://doi.org/10.1145/3746262.3761978>, <https://doi.org/10.1145/3746262.3761978>, Accessed June 26, 2026
64. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 586–595 (2018), <https://api.semanticscholar.org/CorpusID:4766599>, Accessed June 26, 2026
65. Zhang, Z., Fortier-Chouinard, F., Garon, M., Bhattad, A., Lalonde, J.F.: Zero-Comp: Zero-shot object compositing from image intrinsics via diffusion. In: Proceedings of the Winter Conference on Applications of Computer Vision (WACV). pp. 483–494 (February 2025)
66. Zhang, Z., Simo-Serra, E.: G-buffer supported neural screen-space refraction baking for real-time global illumination. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 604–611 (2025). <https://doi.org/10.1109/CVPRW67362.2025.00065>, Accessed June 26, 2026

A Two-step Training and Inference pipeline

We train the conditional one-step diffusion model and the TVAE decoder separately. In Fig. 12 and Fig. 13, we show an overview of each training. Our one-step diffusion model matches the configuration used in SD-Turbo for single-step inference with the img2img pipeline, except that we use the DDIMScheduler [48] rather than the EulerDiscreteScheduler during training and inference. Consequently, our final image is generated from the direct lighting image noised at timestep 499. The frozen VAE decoder in Fig. 12 is replaced with our TVAE decoder (Fig. 13 (green box)) at inference time, as shown in Fig. 14.

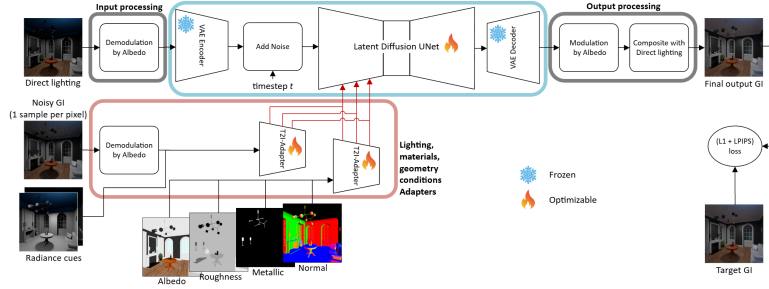


Fig. 12: Overview of fine-tuning pipeline of our conditional one-step diffusion UNet and adapters. We freeze the pre-trained SD-VAE during fine-tuning.

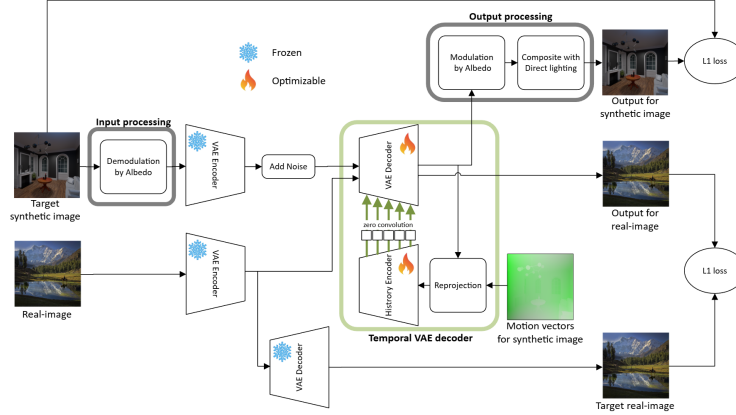


Fig. 13: Overview our temporal VAE training pipeline. As described in Sec. 3.5, we train our TVAE using our synthetic data as well as real-image data, DIV2K [2], in parallel. Reprojection of the previous output using motion vectors is only used for our synthetic data. TVAE is trained with pairs of consecutive frames. Once trained, TVAE decoder (green box) plugs into a diffusion model pipeline by replacing a VAE decoder.

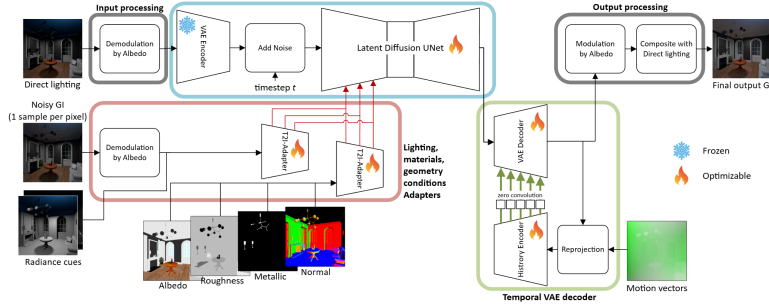


Fig. 14: Inference time pipeline of our method.

B Additional Results

Additional quality metric. CGVQM [26] is a video quality metric that predicts perceptual differences between distorted video and its reference video, taking into account both spatial and temporal artifacts. In Fig. 15, we show error maps generated by CGVQM for our method and three baselines on our test data. The error maps correspond to the same frame shown in Fig. 2, and the metric score overlaid in the bottom-left corner corresponds to the entire video sequence for each test data.

Additional qualitative comparison. We further compare our results against baseline methods on additional test data in Fig. 16 to demonstrate the generalization of our method across diverse scene complexities and lighting conditions. Our method outperforms the baseline methods. Both RGB $\leftrightarrow$ X [62] and DiffusionRenderer [32] exhibit large hallucinations in geometry and materials, with RGB $\leftrightarrow$ X producing results that are much brighter than the reference. BCNN [57] also lacks indirect illumination.

Comparison with a neural denoising method. Since we use a noisy GI image as a sparse lighting condition, our method can be seen as similar to neural Monte-Carlo denoisers. For a wider overview of neural denoising techniques, we refer to a comprehensive survey [24]. We compare our results against a state-of-the-art denoiser, OpenImageDenoise (OIDN) [1]. We used the latest pre-built binary of OIDN (version 2.4.1) and generated results with the highest image quality mode using albedo and normal as clean auxiliary features. In Fig. 17, our results outperform the denoising quality in geometry details, shadows and reflections. Input noisy GI and reference are shown in Fig. 17 (a) and (d), respectively. Note that the scenes shown in Fig. 17 (Factory, Bar) are unseen during training.

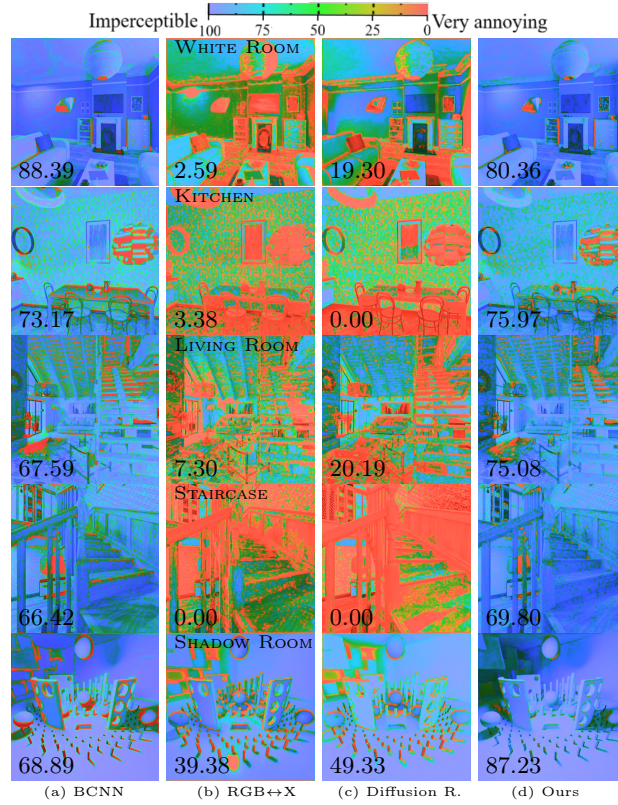


Fig. 15: Visualization of error maps produced by CGVQM [26] for the results shown in Fig. 2. CGVQM $\uparrow$ metric score is shown in the bottom-left corner of each map.



Fig. 16: Additional qualitative comparison between baseline methods and our approach on our test data. Our results outperform baselines in visual quality, better aligning with the reference. We mark with red arrows (→) where baseline methods fail to generate pixels close to the reference.

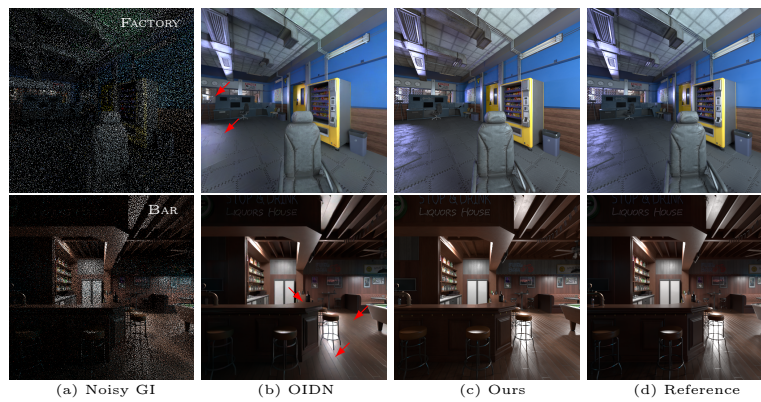


Fig. 17: Qualitative comparison against a neural denoiser, OIDN. Our results outperform in visual quality, better geometry details, shadows and reflections.