TokenLight: Precise Lighting Control in Images using Attribute Tokens

Supplementary Material

In the supplementary, we provide additional implementation details, analysis, qualitative results, and comparisons with baseline methods. Please refer to the supplementary video for a visual overview and further qualitative results.

A. Experimental Setting

A.1. Synthetic Evaluation

Here we provide additional details on synthetic evaluation. This complements our description in the main paper in Sec. 4.1.

For the synthetic benchmark, we compare against two ground-truth targets: PointGT, rendered with the original point light, and PanoGT, rendered using an environment-map representation of that light, for fairness in our comparison with environment-map–based baselines. Here we describe how PanoGT is constructed.

To create the environment map, we replace the point light with an emissive sphere of radius r centered at the original light position. The sphere is assigned a uniform emission L such that its total emitted energy matches the point light [46]:

$$L = \frac{E}{4\pi^2 r^2}. \tag{3}$$

A panoramic camera is then placed at the center of the object to render the environment map, which serves as the lighting input for environment-map baselines. Rendering the object under this map produces the PanoGT target. The panoramic camera captures the incident radiance field $L(\mathbf{x}, \boldsymbol{\omega})$ at location $\mathbf{x} \in \mathbb{R}^3$, where $\boldsymbol{\omega}$ denotes directions on the unit sphere. This provides an approximation to rendering with a 3D point light, as it assumes $L(\mathbf{x}, \boldsymbol{\omega})$ is constant across all surface points in the scene. When the scene is lit by a distant point light, $L(\mathbf{x}, \boldsymbol{\omega})$ varies slowly as $\mathbf{x}$ changes, making this approximation valid. However, the approximation degrades as the point light becomes more local to the scene, where $L(\mathbf{x}, \boldsymbol{\omega})$ exhibits stronger spatial variation. This represents the limitation of using environment map to model spatially varying lighting, and consequently why PanoGT and PointGT exhibit small but non-zero differences, as seen in Tab. 1, last row.

A.2. VisibleFixture-60 Capture Protocol

We capture a dataset of indoor office scenes to evaluate our ability to manipulate lighting from visible light fixtures. We capture a scene with controlled illumination changes by turning the visible light fixture on/off. All captures are performed in office spaces using an iPhone mounted on a tripod, with a fixed camera pose across all shots. We use iPhone's ProRAW mode with auto-exposure enabled. Since the camera remains fixed, no image registration or alignment is required.

The captured ProRAW images are 4K/8K Digital Negative (DNG) images and are converted to EXR format subsequently tone-mapped using the Reinhard operator [49]. For each scene, we manually segment visible light fixtures to identify controllable sources.

B. Additional Implementation Details

Paired Supervision Synthesis Here we provide additional details on synthesizing training pairs, complementing the description in the main paper in Sec. 3.1.

To support intensity, color, spatial, and diffuse controls, we generate supervision on-the-fly during data loading by combining an ambient render I, an on-light render O, and a tone-mapping operator $\mathbf{T}(\cdot)$.

Visible-Light fixtures For visible fixtures, we construct the ambient image by adding the environment-map render with contributions from up to five *non-selected* fixtures. We then choose one fixture to act as the controllable source, use its corresponding render as O, and combine them as

$$I_r = \mathbf{T}\big(a\, I + \lambda\, \mathbf{c}\, O\big), \tag{4}$$

where a is the ambient scale, λ the intensity (both in $[0, 1]$), and $\mathbf{c}$ the sampled color. This setup ensures that only the selected fixture contributes the controllable light, while all remaining illumination is interpreted as ambient, cleanly separating the masked light's effect from the surrounding lighting.

Spatial lighting. For spatial lighting, we follow a parallel construction. The ambient image I is given by the environment-map render, while the controllable source is a point light with a sampled 3D position. The render O corresponds to this single point light, and we again form the relit image as

$$I_r = \mathbf{T}\big(a\, I + \lambda\, \mathbf{c}\, O\big). \tag{5}$$

This formulation provides supervision for precise spatial control: the model learns how a point light at a known 3D location influences the scene.

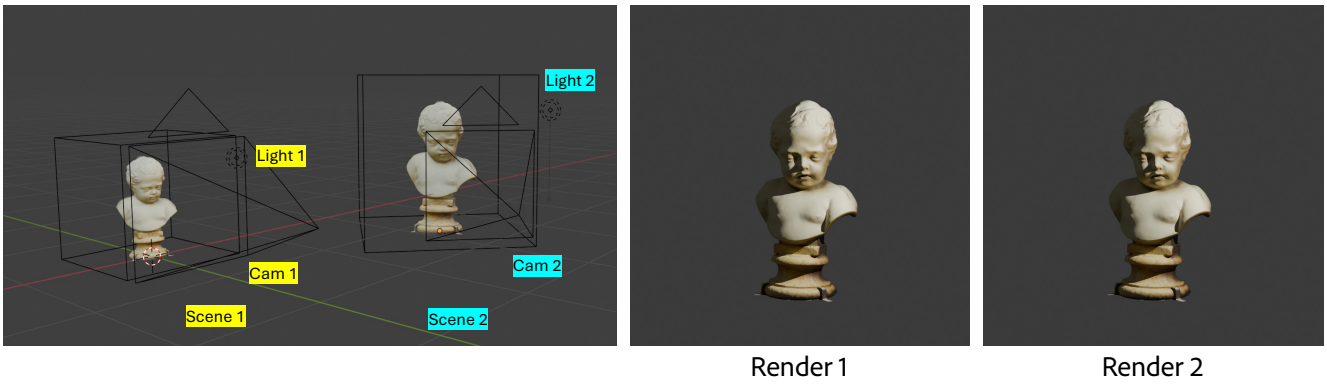


Figure 10. An illustration of the scene-agnostic camera and lighting parameterization, implemented via a transformation from a canonical reference space. Scene 2 is a similarity-transformed version of Scene 1. We map Cam 1 → Cam 2 and Light 1 → Light 2 using the scene transformation defined in our paper. Rendering Scene 1 with Cam 1 produces Render 1, and rendering Scene 2 with Cam 2 produces Render 2, which is visually indistinguishable from Render 1 (only one scene is rendered at a time). This construction allows cameras and lights to be defined in a canonical reference space while scenes are freely placed at different locations, scales, and orientations, ensuring a consistent relationship between lighting parameters and their visual effects in the image space.

Global diffuse-level control For diffuse-spread supervision, we fix the light position and render multiple spread levels. Let A denote the ambient render, O_1 and O_2 the on-light renders with different spread parameters, and I and I_r the corresponding tone-mapped input and target images. Training pairs hold intensity, color, and location constant while varying only the spread parameter:

$$\begin{aligned} I &= \mathbf{T}\big(a\mathbf{c}_1\, A + \lambda\, \mathbf{c}_2\, O_1\big), \\ I_r &= \mathbf{T}\big(a\mathbf{c}_1\, A + \lambda\, \mathbf{c}_2\, O_2\big). \end{aligned} \tag{6}$$

The difference in spread between O_1 and O_2, d_g conditions the model. Note that in order to ensure that perceived differences in shadow softness only stem from O_1 and O_2 we use constant ambient lighting for A, avoiding environment map renders as they can introduce strong directional shadows (e.g., from sunlight) that would appear in both I and I_r, breaking the intended spread-only supervision, since such fixed shadows would confound the effect of changing only the diffuse-spread parameter.

In all three cases, the network predicts I_r, given I and lighting-edit attributes.

Scene-Agnostic Camera and Lighting Here, we provide additional details for our implementation of scene-agnostic camera and lighting that complement the description in the main paper in Sec. 3.2.

We define a canonical reference space containing a camera at position $\mathbf{p}_{\text{cam}}$ and a light sampling volume centered at C. Light parameters, 3D position $\mathbf{p}_{\text{light}}$, energy E, and radius d, are drawn from this space. The cube extends both in front of and behind the image plane, enabling the sampling of lights that fall partially or fully outside the visible frame. All renders use a fixed field of view of $39.6°$ across our synthetic scenes.

During testing, the field of view (FOV) of the input images is unknown. We find that the effectiveness of 3D position controls is not affected, due to the use of normalized coordinates in the reference space.

To expose the model to viewpoint variability while preserving the meaning of the canonical coordinates, we apply 3D similarity transformations to both the lights and the camera during data generation.

Given a target cube center C_t and scale factor s, we first apply a translation that moves both $\mathbf{p}_{\text{cam}}$ and $\mathbf{p}_{\text{light}}$ from the canonical center C to C_t:

$$\mathbf{p}_t = \mathbf{p} + (C_t - C). \tag{7}$$

We then apply a uniform scaling about the new center:

$$\mathbf{p}_{ts} = C_t + s\,(\mathbf{p}_t - C_t). \tag{8}$$

Scaling also adjusts appearance-dependent parameters. The light energy becomes

$$E_s = s^2 E, \tag{9}$$

which preserves perceived brightness by compensating for inverse-square falloff, and the radius becomes

$$d_s = sd, \tag{10}$$

which maintains the angular extent of the light as viewed from the camera. Together, these adjustments ensure that the apparent lighting behavior is invariant under uniform rescaling of the scene.

We optionally apply a rotation $R \in SO(3)$ about C_t:

$$\mathbf{p}_{tsr} = C_t + R(\mathbf{p}_{ts} - C_t). \tag{11}$$

This allows the model to observe the same canonical configuration under diverse orientations while preserving the geometry of camera–light relationships.

In practice, rendering uses the fully transformed parameters $(\mathbf{p}_{tsr,\text{cam}}, \mathbf{p}_{tsr,\text{light}}, E_s, d_s)$, while the model is conditioned only on the *canonical* (pre-transform) parameters $(\mathbf{p}_{\text{light}}, d)$, which are the values provided by users at inference. This separation ensures that tokenized 3D coordinates correspond to a consistent spatial meaning irrespective of the underlying scene content or the particular similarity transform applied during training.

We provide a visualization in Fig. 10 to illustrate how the scene transformation operates. The figure shows two scenes, each containing a camera and a point light, related by a similarity transform. We render both scenes after transforming the light attributes and camera locations under our formulation. The resulting images are visually indistinguishable, confirming that our parameterization preserves lighting behavior under arbitrary scene placement. This enables all lighting parameters to be defined once in a canonical space and ensures that lighting parameters produce consistent visual effects in image space as scenes are placed freely at different locations, scales and orientations.

Additional Training Details We use a 2B-parameter diffusion transformer. Each 960 px image is encoded into a latent of shape $[12, 120, 120]$, patchified 2×2 to $[48, 60, 60]$, projected to 4096-dimensional tokens, and flattened into a 3600-token sequence. Training runs on two nodes (each with 8×A100 80GB) with a total batch size of 160, split as 64:48:48 across visible-light, spatial, and diffuse cases. To enable classifier-free guidance at inference, we drop lighting edit ΔL tokens 10% of the times, replacing them with a tensor of matching sequence length and dimensionality with values -1.

C. Additional User Study Details

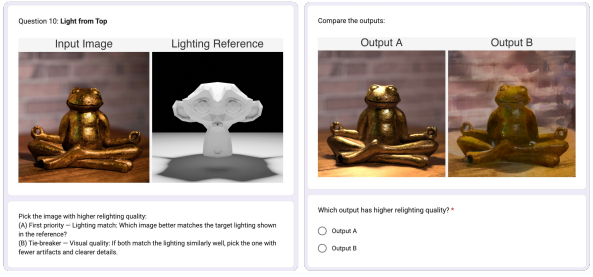


Figure 11. *User Study Questionnaire.*

Here, we provide additional details for the user study summarized in Tab. 2 of the main paper. In the absence of ground-truth lighting, quantitative evaluation on in-the-wild images is challenging. We therefore conduct a user study comparing *TokenLight* against the closest spatial-control baselines, GenLit [3] and Careaga et al. [8]. As shown in Fig. 11, each trial presents an input image, a target-lighting reference, and two relit outputs. Users select the output that better matches the target lighting; if both match the target similarly well, they break ties by choosing the one with fewer artifacts. In the example shown in Fig. 11, the input is a metallic frog figurine, and the target-lighting reference is a Suzanne render lit from above. This reference serves as a visual guide for the lighting cues users should attend to, such as shadow direction and the placement of bright regions. In this example, Output A better matches the intended top-down lighting than Output B and exhibits fewer artifacts. Using 20 in-the-wild images and 5 target lights, we collect votes from 18 users for each baseline comparison. We apply Reinhard auto-exposure [49] to all outputs to reduce perceived brightness as a confounding factor.

D. Effect of Inference steps on Quality/Speed

We evaluate the effect of the number of sampling steps on relighting quality in Fig. 12. We report results using 1, 5, 10, 20, and 50 steps on the synthetic evaluation set described in Sec. 4.1. Each step requires two number-of-function-evaluations (NFE) due to classifier-free guidance. Timing is measured on a single A100 80GB GPU and reflects end-to-end inference time, including pre- and post-processing.

Both LPIPS and SSIM improve with more sampling steps, while PSNR exhibits minor fluctuations but follows an overall upward trend. Qualitatively, plausible lighting emerges even at very few steps, though with visible artifacts (e.g., residual noise patterns at 1 step in Fig. 12(ii), and overly smooth appearance of the pumpkin at low step counts in Fig. 12(iii)) that diminish as the step count increases. This suggests that distillation techniques [57, 68] may be a promising direction for accelerating inference in future work.

E. Additional Qualitative Results

Additional spatial lighting results Fig. 13 presents additional examples of inserting virtual lights at different 3D locations, with emphasis on portrait relighting applications. Our method produces high-quality results that preserve subject identity while robustly handling complex light–geometry interactions.

Spatial lighting under extreme viewpoints Fig. 14 shows results on images captured from top-down camera

viewpoints. These viewpoints are not explicitly seen during training. We evaluate two virtual light trajectories: (a) left-to-right and (b) bottom-to-top, while the light is placed above the objects in both cases. Despite this challenging configuration, our method produces consistent shading and shadows. We attribute this behavior to our camera-agnostic lighting representation and the heavy augmentation of camera viewpoints during training, where we randomize the camera pitch in the range $[-30°, 0°]$.

Additional comparisons with Careaga et al. Fig. 15 provides further qualitative comparisons under additional lighting directions. Our method maintains consistent edits across diverse scenes, while Careaga et al. occasionally introduce unintended appearance changes—such as albedo shifts—likely stemming from intrinsic decomposition errors in their pipeline.

Additional comparisons with GenLit Fig. 16 shows further examples comparing our approach with GenLit. Across these cases, our method has stable light placement and coherent shading, whereas GenLit's outputs often exhibit drift in the intended light position and less consistent illumination effects.

Ambient Lighting Control We provide additional qualitative results demonstrating the range of ambient–illumination edits supported by our model.

Fig. 17 shows continuous scaling of ambient intensity on a variety of examples where a mask is used to specify the light source to be preserved while the remaining light sources, considered ambient lighting, are gradually dimmed.

Additionally, we demonstrate control over shadow softness using the diffuse spread parameter d_g. In Fig. 18, positive values of d_g increase shadow softness by diffusing the light. Since our model is conditioned on the difference in spread parameters, d_g can also be negative. As shown in Fig. 19, negative d_g values reverse the light diffusion process, progressively sharpening shadows in the input image.

Visible-Light Fixture Intensity Control In Fig. 20 and Fig. 21, we demonstrate fine-grained control over visible light-fixture intensities.

In Fig. 20, given an input image and a mask that localizes the visible light source, our method gradually dims the fixture, handling diverse cases including chandeliers.

In Fig. 21, we show the reverse process, where an initially off light is progressively turned on. As intensity increases, the illumination spread follows the geometry of the emitter i.e., the lamp shades and produces plausible, directionally consistent shadows throughout the scene.

Outdoor Visible-Light Fixture Results Fig. 22 shows examples of localized light editing in outdoor scenes. In these results, turning off a car's headlight suppresses only the light emitted by the fixture, while the surrounding illumination from the environment (e.g., sunlight) remains unchanged. Although our method is not explicitly trained on outdoor scenes, our synthetic dataset contains only indoor ones, we render training samples under a variety of environment maps, including outdoor HDRIs. Exposure to such lighting conditions may help the model distinguish localized visible light sources from global environment illumination, enabling plausible behavior in these outdoor examples.

Additional results on VisibleFixture-60 Fig. 23 shows additional results from our VisibleFixture-60 test set, which provides paired captures with visible light sources toggled on and off. Our method turns lights on with illumination that closely matches the reference (row (i)), handles complex disjoint masks involving multiple fixtures (row (ii)), and produces shadows consistent with the captured scene (rows (ii–iii)). Rows (iv–v) show the reverse case: when lights are turned off, associated effects such as shadows (row (iv)) and reflections in the glass pane (row (v)) correctly disappear.

Qualitative comparisons with LightLab LightLab [41] focuses on toggling visible light sources in real indoor scenes. Using author provided test-cases, we perform qualitative comparisons in Fig. 25. Our method reproduces fine-grained effects such as mug shadows and scene reflections (i), and turns lights off while retaining realistic ambient lighting (ii).

F. Independent control of multiple lights

Our representation extends trivially to multiple lights by repeating a compact per-light token block $(\mathbf{p}, \mathbf{c}, \lambda, d)$. Each block parameterizes a single light via its 3D position $\mathbf{p}$, color $\mathbf{c}$, intensity λ, and diffuse level d. We train a model variant supporting up to three lights. During training, we sample $k \in \{1, 2, 3\}$ active lights. For inactive lights, we set their parameters to -1.

Extending the single-light formulation in Sec. 3.1, we construct supervision by summing the contributions of individual lights. Let O_i denote the render corresponding to the i-th light at position $\mathbf{p}_i$ with sampled parameters $(\mathbf{c}_i, \lambda_i)$, and let I denote the ambient render. The relit image is given by

$$I_r = \mathbf{T}\left(a\, I + \sum_{i=1}^{k} \lambda_i\, \mathbf{c}_i\, O_i\right), \tag{12}$$

where $a \in [0, 1]$ is the ambient scale and $\mathbf{T}(\cdot)$ is the tone-mapping operator. This construction mirrors the single-light

case, with the controllable illumination formed by summing per-light contributions.

Since our lighting representation is compact, repeating the per-light token block results in only a modest increase in sequence length; for up to three lights, this incurs negligible additional training and inference cost.

Fig. 26 shows qualitative results with two and three lights. The model produces consistent shading and shadows, with independent light control Fig. 26a and plausible color mixing along shadow boundaries Fig. 26b.

G. Discussion on Video Relighting

Video relighting is a natural extension of our work, reflected by the recent surge of interest [16, 33, 36]. However, extending our lighting representation to video raises interesting research questions, which we discuss here with a focus on spatial lighting control.

The main challenge arises from the fact that the operational space of our spatial lighting representation (i.e., the 3D coordinates exposed to the user) is defined relative to the camera rather than in a canonical 3D world space, raising questions about light placement and persistence as objects/camera move within the video. In the simplest setting, without a driving video, with the goal of animating a relit image, an image-to-video model [61] may be used to propagate lighting edits. Similarly, in settings with a driving video but limited object motion and a static camera (e.g., facial performance capture [23]), our camera-agnostic lighting representation may already be sufficient, provided the model is trained on paired relighting data with the appropriate motion characteristics.

In the general case, where both camera and scene objects move, it remains unclear how a lighting edit specified in the first frame should persist over time. This raises an open question in how to formulate the learning problem—whether via explicit pose tracking, per-frame light tokens, or implicit inference from video. We view this as an interesting direction for future work.

Finally, future work can address autoregressive generation for faster response times. Recent advances [28, 69] in efficient autoregressive video generation suggest a plausible route toward interactive video relighting.

H. Limitations

While our tokenized representation enables intuitive control, several limitations remain. First, our method relies on a large DiT architecture, making real-time interaction for applications requiring immediate feedback currently challenging. Second, the diffusion-based formulation introduces sampling stochasticity—different random seeds produce visually similar but non-identical results (Fig. 24)—most visible at the low-diffuse settings with sharper shadows—which may be problematic for workflows requiring deterministic outputs. Finally, our method generalizes better to indoor scenes than outdoor environments, reflecting a training data bias: our dataset contains no outdoor synthetic scenes. Future work could address these limitations through model distillation for faster inference and training with more diverse outdoor data.

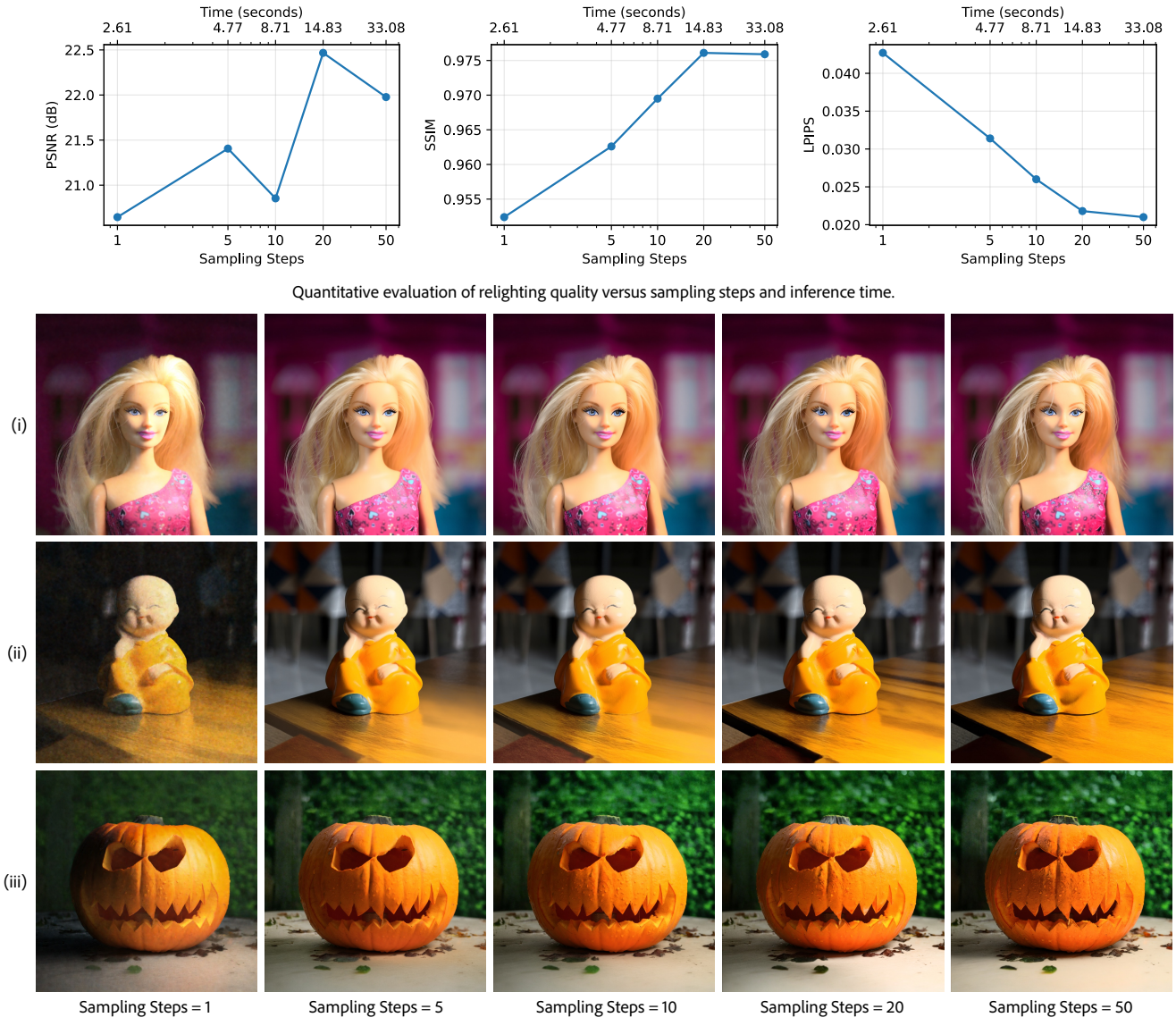
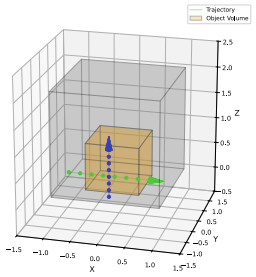


Figure 12. *Effect of sampling steps on relighting quality.* Quantitative evaluation of PSNR, SSIM, LPIPS versus sampling steps and inference time (top). In-the-wild qualitative results at varying step counts for inputs from Fig. 8 rows (i), (iii) and Fig. 9 (i). (bottom).

Figure 13. *Spatial lighting for portraits*: For each input, we show three relit outputs with lights inserted at different 3D locations. The results demonstrate high-quality portrait relighting and complex light–geometry interactions, including challenging cases such as the veiled subject.



(i) Light trajectories

(ii) Top-down input view

(iii) Light moving horizontally (green border) and vertically (blue border)

Figure 14. *Extreme camera viewpoints.* We test on top-down views (left) moving the point light along two trajectories (top). The model follows both motions, left → right (green border) and down → up (blue border), with coherent shading and shadows.

Figure 15. *Additional comparison with Careaga et al. [8]* We show the input image and results for two lighting directions (top-lit and left-lit). In the first row, our method produces plausible relighting on fur. In the second row, it better preserves the material appearance of the input. In the last row, it relights the building while maintaining its albedo.

Figure 16. Additional comparison with GenLit [3] Across these additional examples, our method produces more consistent light placement, while GenLit often exhibits drift in the intended light position.

Figure 17. *Continuous ambient–intensity control.* Each row shows an input image and its light-fixture mask. All remaining illumination is treated as ambient and is progressively reduced by our method. In the first row, for instance, the ambient light from the window fades while the masked light source's reflection in the glass remains unchanged, illustrating proper separation of ambient and the masked light-fixture.

Figure 18. *Ambient-light diffusion for shadow softness*. In each row, we show an input and three results with increasing global diffuse levels that increase shadow softness.

Figure 19. *Ambient-light diffusion for shadow sharpening*. In each row, we show an input and three results with different global diffuse levels. By providing negative d_g values, we can adjust the ambient light diffuse level to make shadows sharper.

Figure 20. *Light-intensity control to gradually turn off lights.* Our method gradually turns off light in input images, provided a mask and different intensity levels, even generalizing to complicated light fixtures such as chandeliers, different types of lamps and streetlights. In the last two rows we show how the mask can be used to localize lighting edits, turning off street lights one at a time.

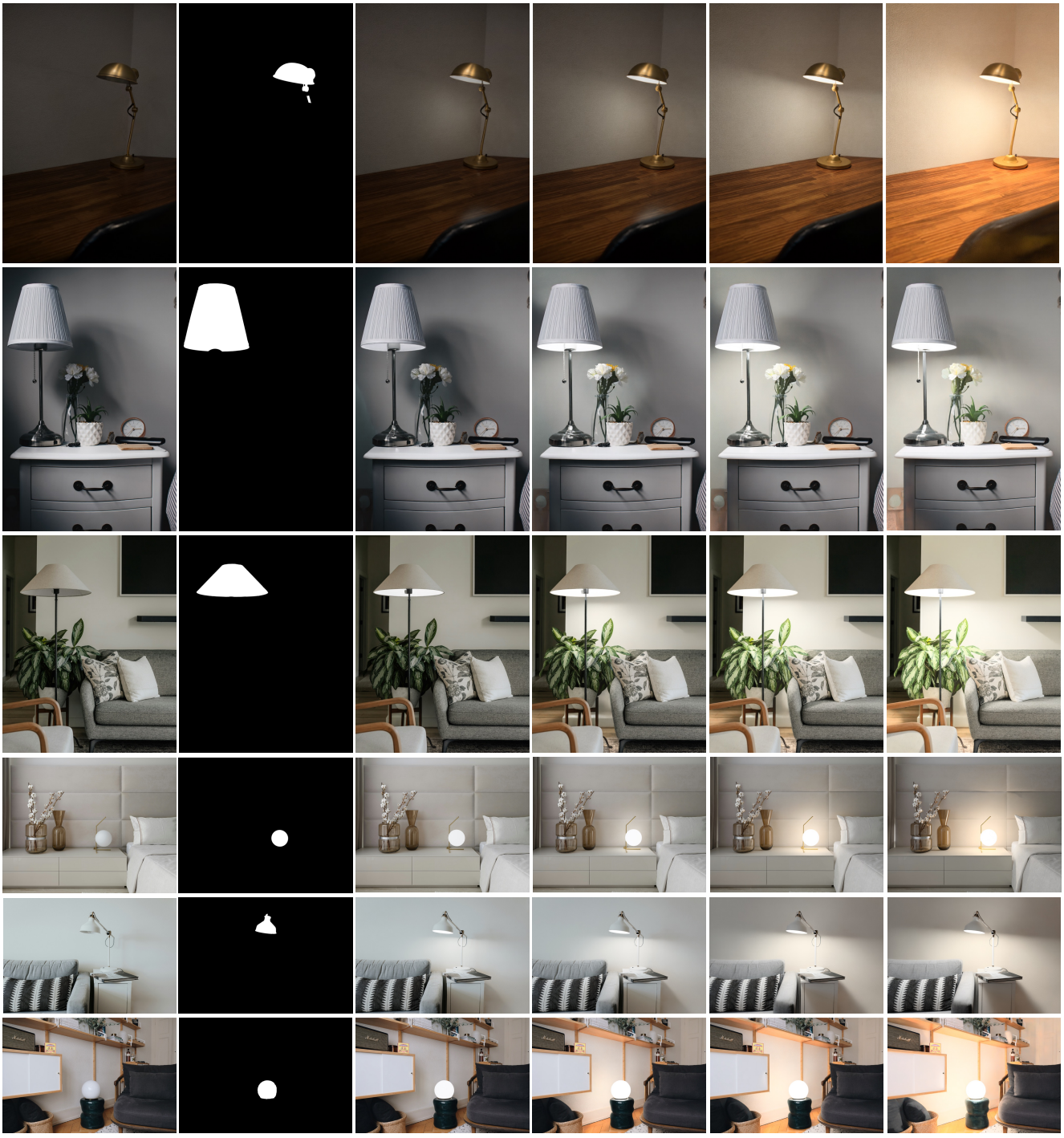


Figure 21. *Light-intensity control for turning lights on*: Given a light-visibility mask, our method increases intensity with realistic light falloffs, i.e., the illuminated portions on nearby walls retain plausible boundary based on light-fixture shape.

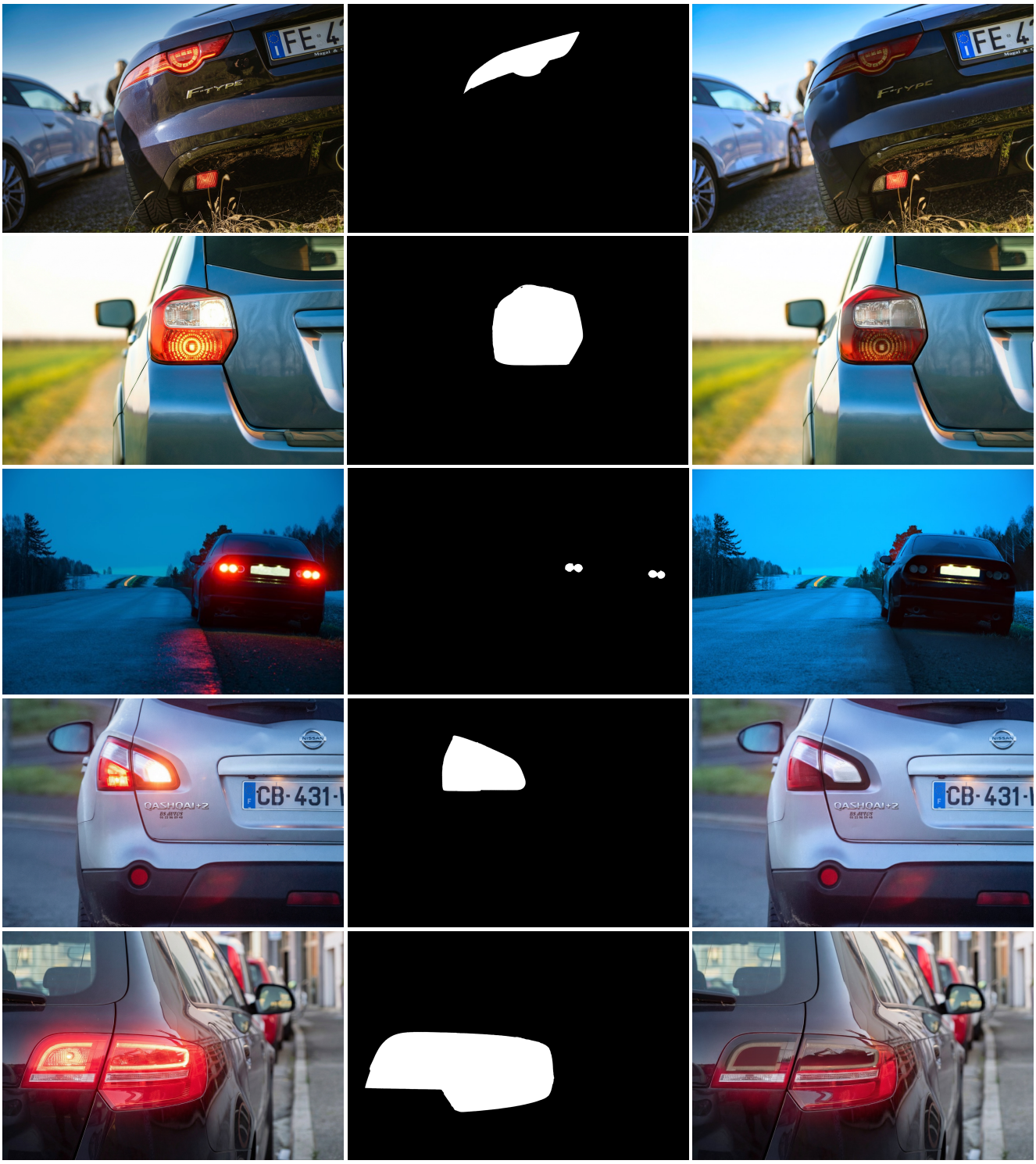


Figure 22. *Outdoor light disentanglement.* Turning off a car's backlight suppresses only the light, without darkening the environment. Our pipeline renders localized light edits under diverse environment maps, promoting this separation.

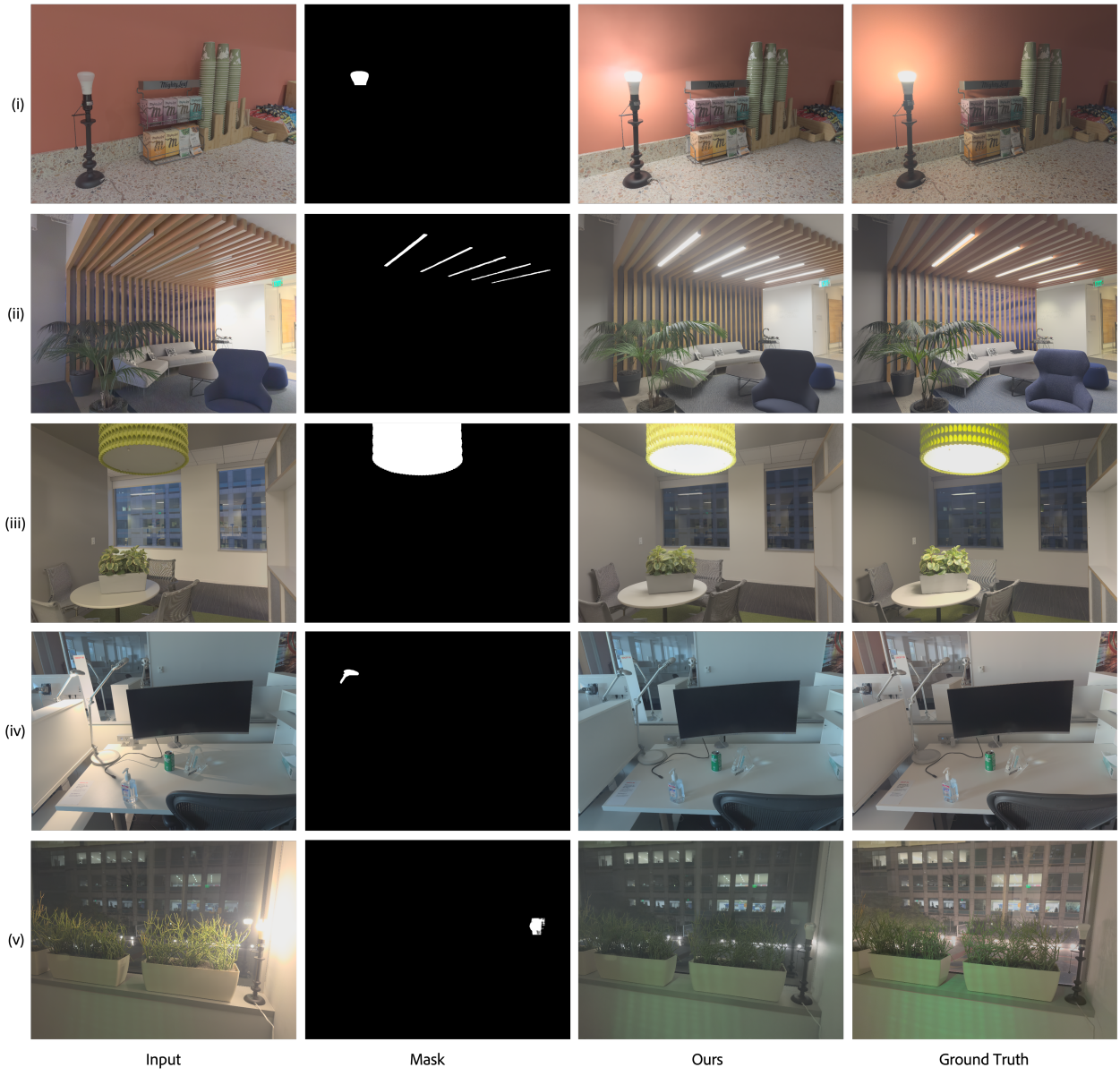


Figure 23. We present additional results on our *VisibleFixture-60* test set with available ground truth. Rows (i-iii) show examples where our method turns lights on while rows (iv-v) show examples where our method turns lights off.

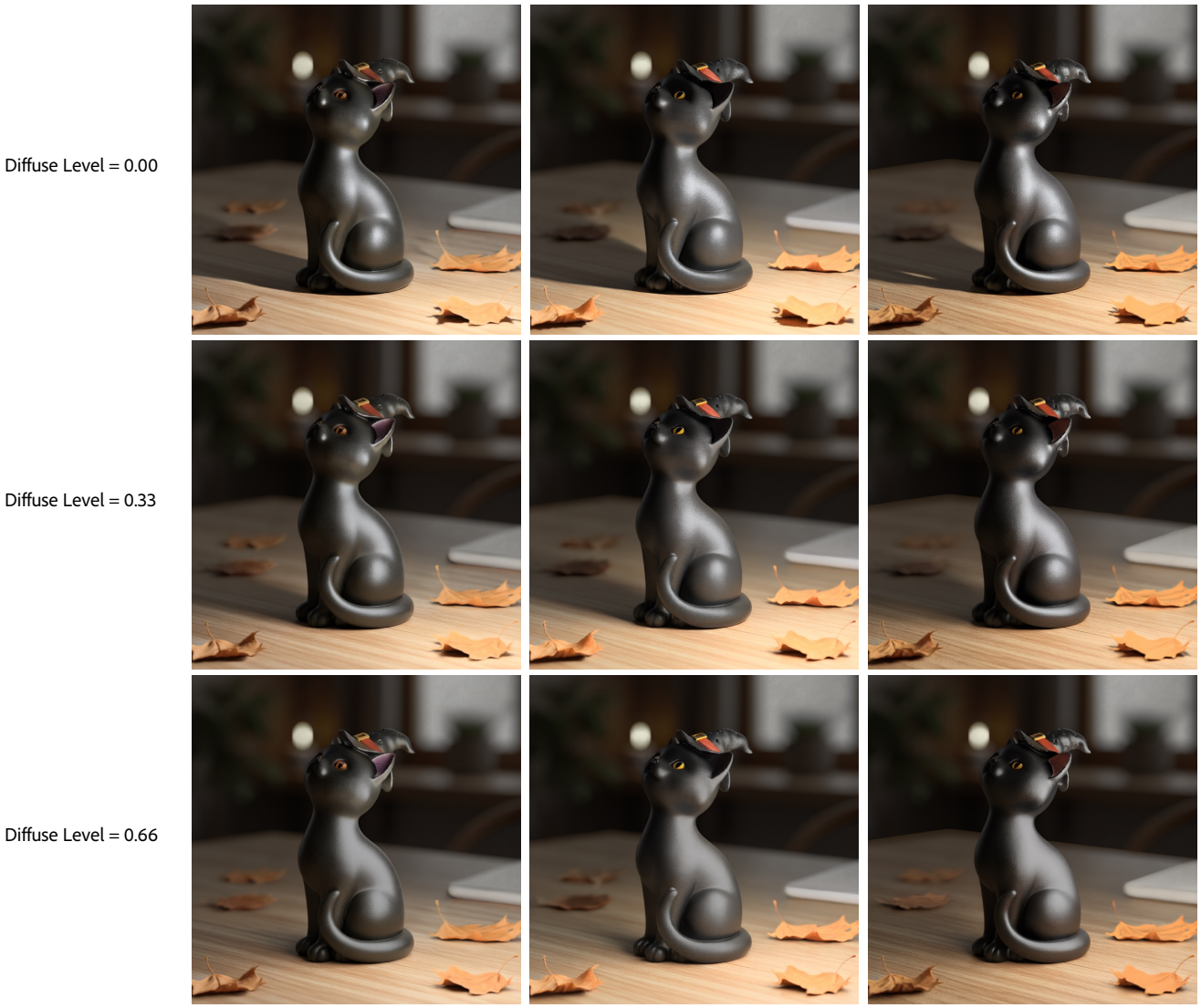
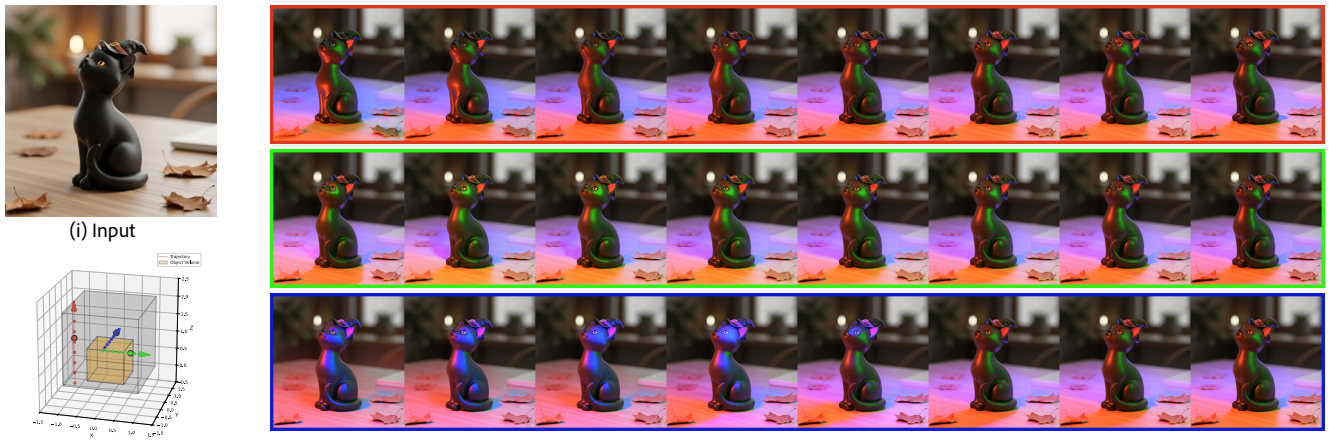
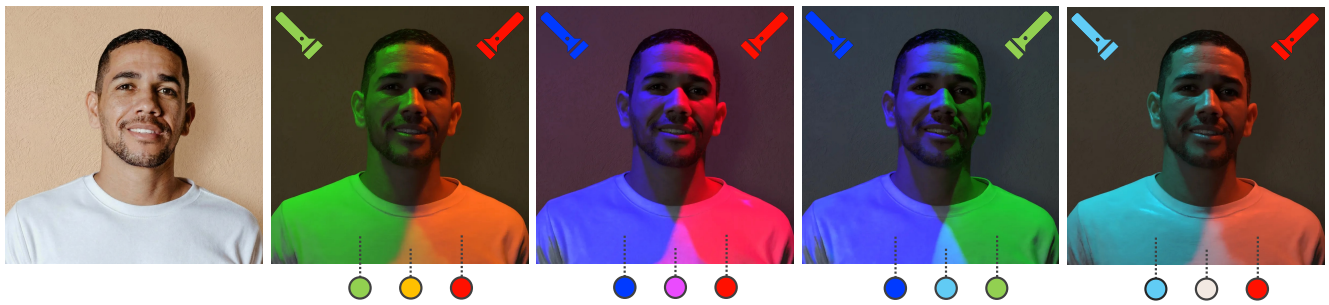


Figure 24. *Seed-dependent variation across diffuse levels.* We insert a virtual light at the same 3D location and vary its diffuse level across the three rows (low to high). Each row shows three outputs generated with different inference seeds. At low diffuse levels, shadows become sharper and exhibit minor seed-dependent shifts. As the diffuse level increases, results become naturally more consistent across seeds. All outputs remain plausible and preserve the intended lighting edit, illustrating that seed-dependent variation is limited to subtle differences.

Figure 25. Comparison with LightLab [41]: Using results provided by the authors, we qualitatively compare our method—reproducing photorealistic effects such as mug shadows and light reflecting from tabletop in (i), and the lamp turning off in (ii).



(a) *Independent multi-light control*. We show the input (top left), the 3D trajectories of three light sources (bottom left), and relighting results obtained by sweeping one light while keeping the other two fixed (right): Top row shows red light moving from bottom to top, middle row shows green light moving from center to right and bottom row shows blue light moving from front to back.



(b) *Color mixing under multiple lights*. We show the input (left) and relit outputs (right) under two fixed light sources placed at the top-left and top-right of the scene. The model produces plausible color mixing and colored shadows, following expected additive behavior across both primary (first three examples) and complementary color combinations (i.e., red + cyan = white).

Figure 26. Our compact light representation can easily be extended to support simultaneous editing of multiple light sources with different attributes (location, intensity, color, diffuse level) within a single inference pass.