Lighting in Motion: Spatiotemporal HDR Lighting Estimation

Christophe Bolduc[1] Julien Philip[2] Li Ma[2] Mingming He[2]
Paul Debevec[2] Jean-François Lalonde[1]

[1]Université Laval, [2]Eyeline Labs

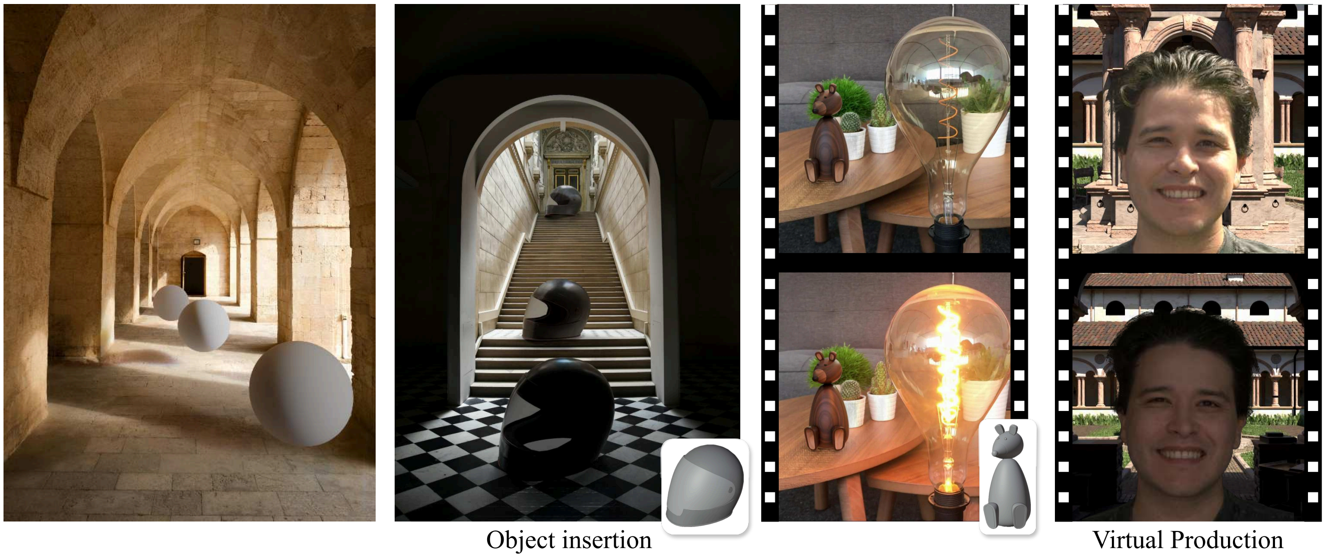

Figure 1. We present LiMo: a spatiotemporal lighting estimation method with accurate spatial grounding, full HDR and realistic reflections. LiMo accurately grounds virtual objects over different spatial positions (1st and 2nd left), and over time (3rd left). LiMo can readily be used in virtual production (right), for instance by inserting actors captured in light domes in real sets.

Abstract

We present ***Li****ghting in* ***Mo****tion (LiMo), a diffusion-based approach to spatiotemporal lighting estimation. LiMo targets both realistic high-frequency detail prediction and accurate illuminance estimation. To account for both, we propose generating a set of mirrored and diffuse spheres at different exposures, based on their 3D positions in the input. Making use of diffusion priors, we fine-tune powerful existing diffusion models on a large-scale customized dataset of indoor and outdoor scenes, paired with spatiotemporal light probes. For accurate spatial conditioning, we demonstrate that depth alone is insufficient and we introduce a new geometric condition to provide the relative position of the scene to the target 3D position. Finally, we combine diffuse and mirror predictions at different exposures into a single HDRI map leveraging differentiable rendering. We thoroughly evaluate our method and design choices to establish LiMo as state-of-the-art for both spatial control and prediction accuracy.*

1. Introduction

Humans have the inherent ability to determine whether a virtual object inserted into an image belongs in the scene or not [14, 38]. When an object's shading does not harmonize well with its surroundings, it creates a "pasted-in" effect where the object appears out of place, breaking realism [32]. Whether the task is to composite an actor into an environment or add an object to an image sequence, having access to accurate lighting information is critical. Capturing lighting has long been the default solution [9], notably by inserting *light probes*—typically, mirror and diffuse spheres reflecting the incoming light rays towards the camera—in the scene and capturing them using high dynamic range (HDR) photography [11]. However, this process requires physical access to the location, the proper equipment to capture HDR images, and time. Hence, the ability for computers to automatically estimate lighting given an image, or a sequence of images, has potential impact for virtual and augmented reality, filmmaking, and design.

We assert that a generally applicable lighting estimation technique should have the five following capabilities:

1. It should allow for "grounding" its prediction in a specific location in the scene, since lighting varies spatially as a function of the relative position to the light sources and occlusions [15]
2. It should adapt to temporal variations: a moving camera revealing unseen light sources, moving objects causing occlusions, or changing lighting conditions
3. It must predict accurate HDR luminance values, including for large areas of indirect light reflecting from objects in the scene as well as for concentrated, orders-of-magnitude brighter light sources
4. It should be able to estimate near-field light sources indoors as well as distant environmental light outside
5. It should estimate plausible lighting distributions including high-frequency environmental detail and low-frequency directional illuminance, even though estimating such information is typically under-constrained

Previous methods focused on subsets of these specifications to break the problem into pieces. For example, methods proposed to estimate a single global lighting estimate from images [15, 23, 30] or videos [28]. Others predict spatially-varying lighting but are targeted for indoors [25, 35] or outdoors [50]. Recent works deal with spatiotemporal variations [26, 41], but we find they struggle in properly grounding the predictions within the local context of the scene.

In this paper, we present what we believe to be the first approach to address all five capabilities in a single framework. Our approach provides lighting predictions that can be grounded at a specific 3D position, vary through time, predict accurate HDR values, works both indoors and outdoors, and generates realistic details for reflections.

Our method, dubbed **Li**ghting in **Mo**tion (LiMo), works as follows: given a monocular image/video and a sequence of positions in the scene, we first use an off-the-shelf predictor [5] to recover per-pixel depth. Using the depth and lighting estimation positions we compute a set of geometric maps that are used to condition a diffusion model. The network, specifically fine-tuned for the task, outputs either a mirror or diffuse sphere at the desired locations, and at a specific exposure value. By querying the network for multiple combinations of mirror/diffuse spheres and exposure values, we obtain a stack of exposure brackets for both the diffuse and mirror spheres at each position. These outputs are subsequently fused for each position into a single HDRI, which are combined into a sequence. To evaluate the method, we present a novel video lighting estimation test dataset made from realistic synthetic data.

Our work makes the following contributions:

- LiMo, a diffusion-based method to predict full HDR lighting at any 3D point in a scene, and at any time in a video.
- New geometric maps to condition the diffusion-based generator, which we demonstrate are critical for accurate spatially-varying predictions.
- A mirror and diffuse multi-bracket approach to lighting estimation which provides both realistic and more physically accurate estimation.

2. Related work

Image-based lighting (IBL)

Classic Image-Based Lighting photographs light probes, typically mirror and diffuse spheres [9, 33], to construct HDRI maps and render realistic virtual objects. Hardware and physical constraints prohibit the use of such techniques if physical access to the scene is impossible, and as such, methods for estimating lighting from images have been proposed.

Single image lighting estimation

Early approaches estimated environment lighting from images using cues such as reflections, shadows, and geometry [22, 34]. Of course, learning-based methods outperformed their predecessors by proposing approaches that directly regress HDR lighting representations from single images of indoor scenes [15, 16, 46], outdoor scenes [20, 47], or both [6, 8, 23, 30], or a scene with a human face [24]. The interested reader is invited to consult the survey of Einabadi *et al.* [13] for more details. The aforementioned approaches typically estimate lighting either at the center of the image or at a single point in the scene.

Spatially-varying lighting estimation

Since lighting can vary drastically across the field of view, some methods accept a specific location as input, and predict the lighting at that point as an HDRI map [1, 35] or spherical harmonics [17]. Other methods predict a dense lighting representation, for example at each pixel location in a 2D grid of spherical gaussians [25, 49] or as a volumetric, either using a voxel grid of spherical gaussians [44], or an implicit representation [36]. Other spatially-varying approaches have been proposed for outdoor scenes [39, 50].

Spatiotemporal lighting estimation

Lighting can also change over time: flipping a light switch, or panning the camera to a bright window, create drastic changes in lighting that can be estimated by constructing a spatiotemporal volume [26]. Concurrent to this work, LuxDiT [28] predict temporally-varying HDRI maps. Finally, and most closely related to our work, Tong *et al.* [41] adapt a diffusion model for generating multiple spheres across the field of view, and subsequently build a unique implicit representation over the video through a NeRF-like approach. However, as we will demonstrate, it tends to

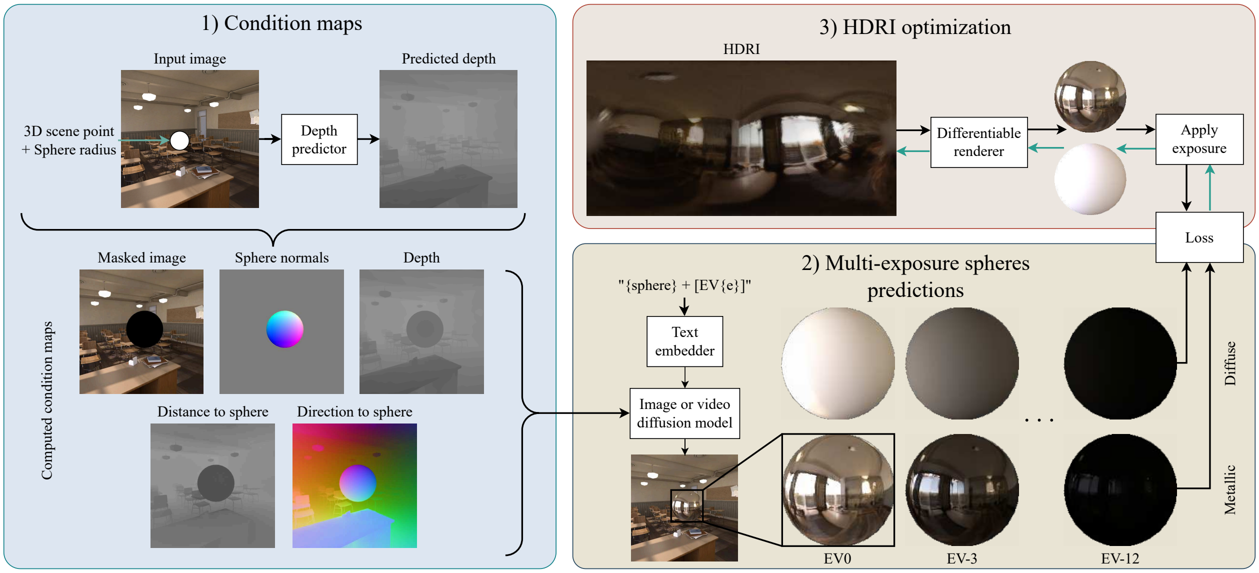

Figure 2. Overview of LiMo, our proposed diffusion-based spatiotemporal lighting estimation method. From an input image (or video sequence) and a 3D scene point, we first obtain an estimate of the per-pixel depth (1, top). From this, a set of condition maps are computed (1, bottom). These maps, along with a text prompt, are used to condition a diffusion model (2) which is trained to inpaint a sphere at the desired 3D scene point. The model learns to predict spheres at different exposures and materials (metallic or diffuse). The predicted spheres are merged into a single HDRI map (3) through a differentiable rendering approach.

generate results that are overly smooth and do not properly capture the dynamics of lighting.

Diffusion-based rendering

It has been recently shown that virtual object insertion can be performed as a learning task, through diffusion-based rendering [27, 45, 48] or harmonization [4, 19], bypassing the need for acquiring or estimating an HDR map. While these approaches offer a promising new paradigm for image compositing, they lack the artistic control offered by the traditional IBL pipeline. Here, we focus on explicit HDR lighting recovery in the form of HDRI maps, as they can readily be used in existing, production-ready object compositing frameworks.

3. Method

We present LiMo, which adapts image and video generative models to predict spatially-varying HDR illumination in a single image, or a video sequence.

3.1. General approach

Our approach consists of using priors from a diffusion model to inpaint a diffuse or mirror sphere at a specific 3D position in space, and at a given exposure value. At test time, the model is run multiple times to generate both diffuse and mirror spheres at multiple exposures, and predictions are subsequently merged to a single HDRI.

Contrary to some recent work [6, 30] that only rely on mirror spheres, we argue that accurate HDR lighting estimation is greatly facilitated by using diffuse spheres. Mirror spheres are a great way to obtain plausible reflections but the estimation of luminance values for concentrated light sources relies on a very small number of pixels and requires many exposure values [37]. With a diffuse sphere, the energy of concentrated light sources is integrated to moderate exposure levels by the diffuse surface and becomes easier to estimate accurately from fewer exposure levels [10]. Figure 2 shows an overview of our proposed method. Next, we elaborate on how we condition the diffusion model (Sec. 3.2), the dataset used for fine-tuning it (Sec. 3.3), and the HDRI reconstruction at test time (Sec. 3.4).

3.2. Model conditioning

We fine-tune a diffusion model, conditioning it on an input image or video, additional maps, and text. For clarity, we describe what is done for single images, but our approach handles videos by using a video model or applying the image model independently on each image sequentially.

RGB and geometry conditioning To allow accurate lighting estimation, we curate multiple input maps used as conditioning to the diffusion model. These input maps are channel-wise concatenated to the input noise and the first layer of the model is expanded to accommodate these extra channels.

First, we provide the RGB image as input I_{rgb}. The region corresponding to the sphere to inpaint is set to black to prevent the background from spilling into the sphere. Second, we provide the depth of the background image I_{d}, estimated from an off-the-shelf depth predictor [5], with the depth of the sphere to inpaint.

As we will later demonstrate (see Sec. 4.5), we found that providing only the depth of the scene and sphere, as in [41], is insufficient for accurately grounding the light prediction at a specific 3D point in the scene. We therefore provide three additional maps: a normal map of the sphere I_n, and two novel maps capturing the geometric relations, providing context between the scene and the sphere.

Those geometric maps relate the scene's surfaces to the sphere's position: for a given pixel i in the image, we define the direction $I_{\text{dir},i}$ and distance $I_{\text{dist},i}$ from the 3D point corresponding to i to the sphere as

$$I_{\text{dir},i} = \begin{cases} \dfrac{\mathbf{p}_i - \mathbf{c}}{\|\mathbf{p}_i - \mathbf{c}\|}, & \text{if not on sphere,} \\ \mathbf{v}_i - 2(\mathbf{v}_i \cdot \mathbf{n}_i)\,\mathbf{n}_i, & \text{if on sphere,} \end{cases} \tag{1}$$

and

$$I_{\text{dist},i} = \|\mathbf{p}_i - \mathbf{c}_i\|\,. \tag{2}$$

Here, $\mathbf{p}_i$ are the (x, y, z) pixel world coordinates, computed from the pixel depth and view direction vector $\mathbf{v}_i$ obtained using the estimated camera field of view from an off-the-shelf method [43], and $\mathbf{c}$ is the coordinates of the sphere center. For pixels on the sphere, we use the reflected incoming ray direction, computed using $\mathbf{n}_i$, the sphere surface normal and $\mathbf{v}_i$. Intuitively, this map allows the model to match the reflected direction at a given pixel on the sphere and points in the scene that are in the same direction from the sphere center. These geometric quantities are expressed in the camera coordinate system.

The image I_rgb is assumed to be in sRGB color space, and both the depth map I_d and distance to sphere map I_dist are log-normalized. All maps $\{I_\text{rgb}, I_\text{n}, I_\text{d}, I_\text{dir}, I_\text{dist}\}$ are individually encoded to latent space using the pre-trained VAE encoder, and subsequently channel-wise concatenated. To allow more channels, the patch embedding convolution of the denoising network is adapted by copying and dividing the initial pre-trained weights by the number of added channels.

At training time, the ground truth per-pixel depth I_d and camera field of view are used to compute the maps. At test time, we rely on pre-trained estimators (Chen et al. [5] for depth and Wang et al. [43] for field of view).

Exposure conditioning Similar to Bolduc et al. [3], we train the model to accept an exposure value (EV) as text prompt, given to the network using the pre-trained text encoder. Contrary to methods which interpolate between 2 extreme exposures [30, 41], we found that giving a discrete number of EVs is effective for accurate EV predictions. In practice, we set EVs to $\{0, -3, -6, -9, -12\}$. During training, the corresponding EV image sphere is given as target, keeping regions outside the sphere at the original EV0.

Multi-sphere predictions An additional text prompt is given to condition on which sphere to inpaint (mirror or diffuse). Including both the type of sphere and the exposure value, the final text prompt has the form "{sphere type} [EVvalue]" (e.g., "Diffuse sphere [EV0]").

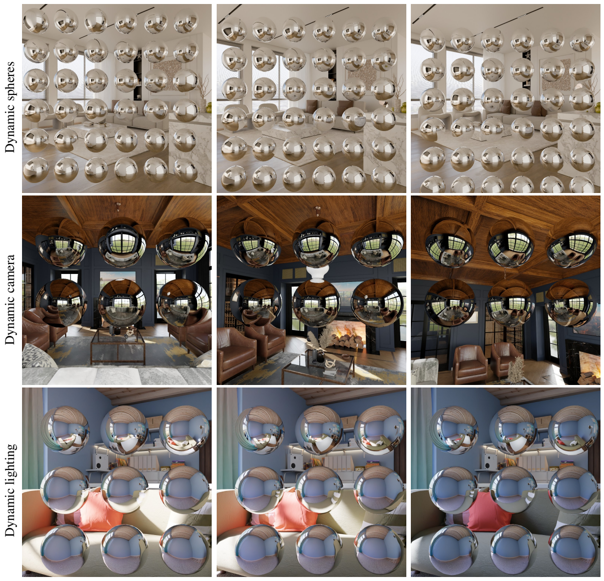

Figure 3. Sample frames from the training dataset, illustrating the three animation scenarios. In addition to the mirror spheres, the diffuse spheres and the empty scene are also rendered (not shown). Note that, while multiple spheres are rendered simultaneously for more efficient data generation, at training and inference the network regresses only one sphere at a time.

3.3. Dataset

To train our approach, physically acquiring a dataset of ground truth HDR lighting moving in space in dynamic lighting would require a highly controllable environment. Instead, we turn to synthetic scenes to generate sequences and their associated ground truth illumination. We use Blender [7], paired with BlenderKit [12] assets to procedurally generate both indoor and outdoor renderings. More details on this process are available in the supplementary materials.

To render a data sample, we randomly select a scene (indoor or outdoor), select a random camera pose, and first obtain a normal RGB rendering. We then generate training targets by placing a grid of spheres in the scene, disabling their visibility to both other spheres and the scene. They are equally-spaced and sized in image space and of random depth. Even though our training is done on a single sphere, packing a sphere grid provides multiple data points per render and viewpoints, allowing for a random sphere selection as training sample. Each sphere depth is computed using the following factor :

$$\delta = d_\text{min} + (d_\text{max} - d_\text{min})u^\alpha, \tag{3}$$

with $d_{\text{min}} = 0.25$, $d_{\text{max}} = 0.98$, $\alpha = 0.4$ and $u \sim \mathcal{U}(0, 1)$. To obtain the scene scale depth for each sphere, the sampled depth factor is multiplied by the scene's minimum depth over the area of the sphere:

$$d_{\text{sph}} = \delta \min(I_d \odot M_{\text{sph}}), \tag{4}$$

where I_d is the first render's depth and M_{sph} is a binary mask of the sphere's footprint in image space. Finally, the 3D radius is computed according to the sampled depth and camera's field of view to allow fixed image space radius.

The sphere is rendered with two materials: perfect mirror (*roughness* = 0, *metallic* = 1, *albedo* = (1,1,1) and perfect diffuse (*roughness* = 1, *metallic* = 0, *albedo* = (1,1,1)). Images are rendered as float16 EXR, retaining the high dynamic range. We also save the depth and sphere masks layers, along with the intrinsic camera matrix and the sphere's 3D position and world radius. These extra layers enable the computation of all our condition maps.

For video sequence data generation, the approach is adapted to 3 scenarios, illustrated in Fig. 3.

Dynamic sphere position In this scenario the camera is static and the spheres are moving. A random image space offset (x', y') is selected $x' \sim \mathcal{U}(0, W)$, $y' \sim \mathcal{U}(0, H)$, where W and H are the image width and height respectively. The spheres will move by this offset in image space over the sequence. For depth, an additional factor is sampled for the last frame and the 3D position and world size of the sphere is interpolated through the sequence.

Dynamic camera We procedurally generate two camera poses in the scene and interpolate between them. The absolute distance to the sphere is interpolated between the first and last frame with a fixed depth factor to insure smoothness of movement.

Dynamic lighting The lighting is made dynamic by randomly rotating the azimuth of the HDRI map, randomly changing the scene's light source intensities, and randomly rotating the "Sun" light.

3.4. Equirectangular HDRI map optimization

At inference, the diffusion model is queried multiple times to generate both mirrored and diffuse spheres at different exposures. We obtain a final HDRI by merging the predictions.

For this, we define a rendering function $\mathcal{R}(L_t, m)$, which renders a sphere of material $m \in \mathcal{M}$, with $\mathcal{M} = \{\text{mirror}, \text{diffuse}\}$, under HDRI lighting L_t. We then seek to find

$$\arg\min_{L} \sum_{t \in \mathcal{T}} \sum_{e \in \mathcal{E}} \sum_{m \in \mathcal{M}} \ell\left(\pi(e, m, t) - e\mathcal{R}(L_t, m)\right) , \tag{5}$$

where $\pi(e, m, t)$ is the fine-tuned generative model conditioned on frame t, exposure $e \in \mathcal{E}$, where $\mathcal{E} = 2^{\{\text{EV}_0, \text{EV}_{-3}, \ldots\}}$ and material m, and ℓ is the loss function:

$$\ell = \ell_2(\hat{y}_t, y_t) + \frac{\lambda}{2}(\ell_1(\hat{y}_t, \hat{y}_{t-1}) + \ell_1(\hat{y}_t, \hat{y}_{t+1})). \tag{6}$$

L is initialized at a constant gray value (0.5), and Adam is used to optimize Eq. (5) through gradient descent, as we employ differentiable rendering to implement $\mathcal{R}$. The latter is implemented in PyTorch as a two modes renderer: reflective and diffuse with cosine and light multi-importance sampling. In practice, we randomly alternate between exposures and materials at each iteration instead of summing over all possibilities. To speedup convergence, we also employ a Laplacian pyramid representation for L [18].

4. Experiments

4.1. Implementation details

We train two models: an image diffusion model and a video diffusion model. We fine-tune the full image Flux.1 Schnell model [21] for 150k steps using 12896 images at 512 × 512 resolution. For the video model, we use Wan2.2 5B [42] model and fine-tune for 250k steps using 30096 sequences of 21 frames also at 512 × 512 resolution. For both versions, a series of color, exposure and degradation augmentations is employed. The training takes 50 hours for the image model and 188 hours for the video model on 8 A100E GPUs.

4.2. Evaluation metrics, datasets and baselines

As is typical for lighting estimation methods, we compute metrics on spheres relit with the predicted HDRI maps. This allows comparing both the specular appearance and physical accuracy of the HDRI on different materials. Concretely, we render spheres of different materials with the predicted HDR illumination: a perfect mirror, perfect diffuse, semi-rough metallic (herein called matte), and perfectly glossy. We evaluate our method for single image predictions on a synthetic and a real dataset. The synthetic dataset consists of 28 scenes from Infinigen Indoor [31], in which 4 light probes are randomly scattered in space. The Laval Indoor Spatially Varying HDR dataset [17] is a real dataset of physical probes placed in a scene and captured in HDR. Because mirror light probes are not perfectly reflective, the authors of the original dataset graciously shared the reflectivity of the sphere used (74%), which we used to adjust the HDR probes and treated as ground truth. For video predictions, we take 5 of the Blender demo files [2] and augment them by animating the cameras, moving probes and modifying the light sources. We compare our method against DiffusionLight [30] and 4D Lighting [41] using their public implementations.

We report RMSE to inform the intensity of the predicted illuminance, SI-RMSE and SSIM as indicators of the structure of the predictions, and RGB angular error to assess the color reconstruction. For temporal results, as is typical of video lighting tasks [29], we report T-LPIPS, LPIPS on neighboring frames. However, to account for the motion of the ground truth scene, we also report T-LPIPS-Diff, the absolute difference between T-LPIPS of the prediction and

		RMSE$_\downarrow$				SI-RMSE$_\downarrow$				SSIM$_\uparrow$				Ang. Err.$_\downarrow$			
Dataset	Method	Mirr	Diff	Gloss	Mat	Mirr	Diff	Gloss	Mat	Mirr	Diff	Gloss	Mat	Mirr	Diff	Gloss	Matte
Infinigen	Diff.Light	0.40	0.47	0.48	0.43	1.52	0.70	0.70	0.85	0.68	0.83	0.81	0.83	14.3	9.7	9.4	9.8
	4D Lighting	0.34	0.36	0.38	0.38	1.36	0.62	0.64	0.74	0.72	0.86	0.85	0.84	14.7	11.2	11.5	11.6
	LiMo (image)	0.25	0.16	0.16	0.17	0.41	0.11	0.13	0.18	0.78	0.95	0.94	0.95	4.4	2.3	2.2	2.4
	LiMo (video)	0.26	0.22	0.22	0.21	0.42	0.13	0.16	0.21	0.79	0.95	0.93	0.94	4.4	2.7	2.8	2.9
Laval Indoor SV	Diff.Light	0.50	0.49	0.51	0.49	1.40	0.91	0.89	0.99	0.70	0.80	0.78	0.80	10.3	8.2	8.2	7.8
	4D Lighting	0.35	0.27	0.28	0.31	0.91	0.22	0.23	0.39	0.80	0.94	0.93	0.92	6.8	5.0	5.0	5.1
	LiMo (image)	0.30	0.20	0.22	0.24	0.60	0.17	0.18	0.28	0.81	0.97	0.95	0.95	4.6	2.7	2.7	2.9
	LiMo (video)	0.35	0.27	0.28	0.30	0.66	0.21	0.23	0.34	0.80	0.94	0.93	0.93	5.5	3.7	3.6	3.9

Table 1. Quantitative evaluation of lighting estimation on single images from the Infinigen [31] (top) and the Laval Indoor SV datasets [17] (bottom). We compare the image and video versions of LiMo with "Diff.Light" [30] and "4D Lighting" [41]. "Mirr" (mirror), "Diff" (diffuse), "Gloss" (glossy) and "Mat" (matte) refer to the different test spheres (see Sec. 4.2). Results are color coded by best, second and third best. We observe that LiMo (image) outperforms the previous work in all cases.

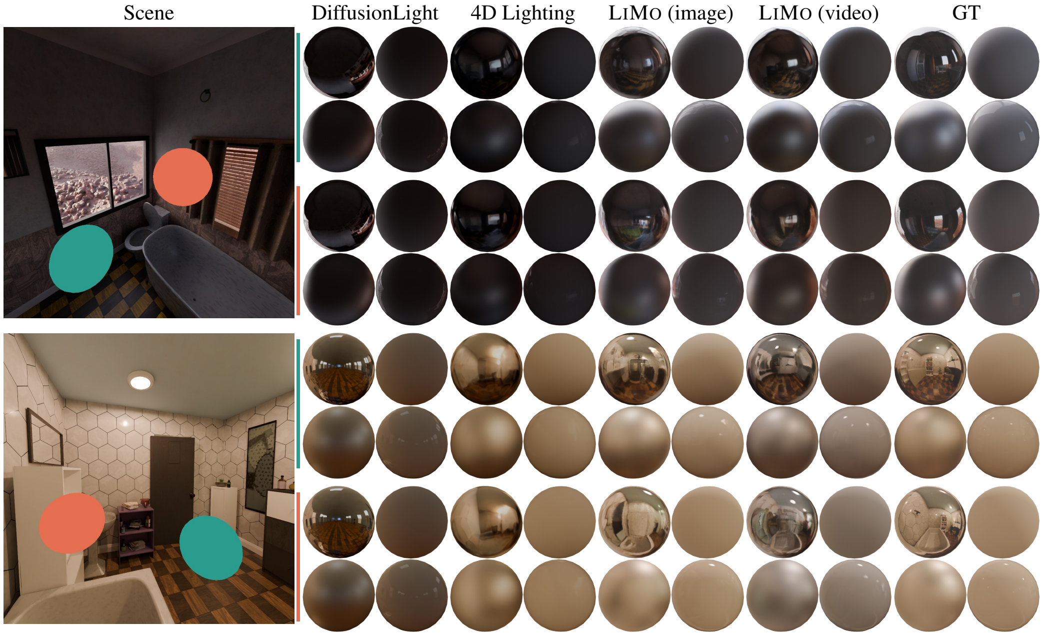

Figure 4. Sample predictions from the Infinigen test set [31] for, from left to right: DiffusionLight [30], 4D Lighting [41], and the image and video versions of the proposed LiMo. We visualize predictions by rendering the same four test spheres used for the quantitative metrics (see Tab. 1): mirror (top left), diffuse (top right), matte (bottom left) and glossy (bottom right).

the ground truth. In addition, we report the warped error, computed by warping the current frame of the prediction by the optical flow predicted from an off-the-shelf module [40] and comparing with the RMSE of the next frame.

4.3. Single image results

Quantitative results for single image predictions are provided in Tab. 1. On both synthetic and real data, our image model performs better than the two baselines, with our video model trailing close behind. We attribute the discrepancy between both versions to the capacity of the respective models (12B parameters for image vs 5B parameters for video) and to the optimization for both quality and time consistency of the video model. The metrics from 4D Lighting for Infinigen differ from the numbers reported in Tong et al. [41] since we sample a new set of scenes and spheres. The metrics confirm that our method is better for predicting total illuminance (from the RMSE), accurate geometry (from SI-RMSE and SSIM), and better colors (from angular error). DiffusionLight, predicting a single HDRI for each scene, cannot adapt to the 3D locations of the spheres' placements, as reflected in the lower scores. As for real scenes, our method achieves better or equal (for video) results than 4D Lighting, showing strong generalization to real images.

		RMSE$_\downarrow$		SI-RMSE$_\downarrow$		SSIM$_\uparrow$		Ang. Err.$_\downarrow$		T-LPIPS$_\downarrow$		T-LPIPS-Diff$_\downarrow$		Warped Err$_\downarrow$	
Dataset	Method	Mirr	Diff	Mirr	Diff	Mirr	Diff	Mirr	Diff	Mirr	Diff	Mirr	Diff	Mirr	Diff
Dynamic object	4D Lighting	0.39	0.29	1.18	0.20	0.70	0.90	7.1	4.7	0.0048	0.0004	0.0418	0.0009	0.0439	0.0079
	LiMo (image)	0.28	0.15	0.43	0.10	0.77	0.97	3.0	1.4	0.1340	0.0054	0.0886	0.0045	0.1887	0.0330
	LiMo (video)	0.30	0.18	0.45	0.12	0.78	0.97	4.5	3.0	0.0242	0.0014	0.0227	0.0013	0.0589	0.0134
Dynamic camera	4D Lighting	0.39	0.37	0.88	0.17	0.71	0.90	6.5	3.7	0.0057	0.0007	0.0279	0.0007	0.0354	0.0124
	LiMo (image)	0.30	0.16	0.44	0.10	0.76	0.97	3.3	1.4	0.1051	0.0057	0.0715	0.0053	0.1500	0.0394
	LiMo (video)	0.30	0.23	0.47	0.12	0.77	0.97	4.4	2.5	0.0220	0.0010	0.0148	0.0011	0.0506	0.0122
Dynamic lighting	4D Lighting	0.39	0.33	1.37	0.69	0.68	0.88	12.6	10.0	0.0030	0.0006	0.0067	0.0005	0.0158	0.0093
	LiMo (image)	0.28	0.16	0.44	0.11	0.78	0.97	3.6	1.8	0.0162	0.0018	0.0085	0.0012	0.0336	0.0193
	LiMo (video)	0.34	0.22	0.49	0.14	0.76	0.96	4.7	2.9	0.0027	0.0008	0.0065	0.0005	0.0108	0.0076
Combination	4D Lighting	0.38	0.31	0.98	0.19	0.70	0.91	7.7	3.9	0.0071	0.0010	0.0496	0.0013	0.0558	0.0150
	LiMo (image)	0.29	0.16	0.46	0.11	0.77	0.97	3.7	1.9	0.1354	0.0075	0.0787	0.0060	0.1941	0.0463
	LiMo (video)	0.33	0.23	0.48	0.15	0.77	0.95	4.6	2.7	0.0370	0.0031	0.0208	0.0030	0.0780	0.0250

Table 2. Quantitative evaluation of lighting estimation on dynamic scenes. We compare LiMo with "4D Lighting" [41]. "Mirr" (mirror) and "Diff" (diffuse) refer to the different test spheres (see Sec. 4.2). Due to space limits, Glossy and Matte metrics are omitted and available in the supplementary materials. Results are color coded by best, second best.

Visual samples from the predictions, shown in Fig. 4, visually demonstrate the higher quality reflections in comparison to 4D Lighting and the better HDR predictions, particularly when looking at the glossy highlights.

4.4. Video results

To evaluate the temporal results of our method, we devise four test cases: dynamic object, dynamic camera, dynamic lighting and a combination of the above. Our novel test set, based on 5 augmented Blender demo files, is used for evaluation. The scores presented in Tab. 2 tell a similar story for the metrics per-frame, where our image model outperforms 4D Lighting, with our video model close second. However, for the three temporal metrics, the video model beats the image model. T-LPIPS, a typical measurement of lighting consistency, is missleading as a certain amount of motion is expected. To compensate, the T-LPIPS-Diff metric compares the T-LPIPS of the prediction to that of the ground truth. Here we see that in every mirror render, 4D Lighting does not vary as much as it should, and our video model is equal or close behind for diffuse renderings. Although we observe lower warped L2 error metrics with 4D Lighting for some experiments, we attribute them to over-smoothing from the MLP formulation. This can be seen in our lower temporal metrics for the lighting change scenario where abrupt discontinuities are required. Moreover, to demonstrate the capabilities of LiMo, Fig. 5 shows samples from the test dataset. Of note is the inability of 4D Lighting to vary the lighting appropriately as the sphere is pushed farther into the scene, whereas ours warps as is expected of 3D space. More in-the-wild results can be found in Fig. 1 and the supplementary materials.

4.5. Ablations

To justify our design choices, we ablate the use of the diffuse sphere for better HDRI predictions and the use of our novel geometric maps as opposed to the depth maps uniquely. All ablations are performed with the LiMo (image) model on the Infinigen test set. Metrics are reported in Tab. 3, where we observe that our full model performs better in every scenario.

Predicting the diffuse sphere The metrics validate the effectiveness of using the predicted Diffuse sphere in conjunction with the mirror sphere for the optimization of the final HDRI. Notably, the color prediction, as informed from the angular error, is worse when the diffuse sphere is omitted.

Geometric maps As discussed in Sec. 3.2, we observed that depth maps of the spatial position are insufficient for the network to properly inpaint a correctly-placed sphere. This observation is illustrated in Fig. 6, where a sphere is kept the same image space size, but pushed farther into the scene, from shadow to direct sunlight. In the case of depth conditioning only, the two inpainted spheres are nearly identical. However, when introducing our novel geometric maps, the network is able to interpret the position of the sphere in relation to the scene's elements and correctly inpaint a shadowed and lit sphere respectively. Those results are validated by the metrics, where removing the geometric maps results in worse performance than removing the diffuse sphere. Interestingly, removing both the geometric maps and the diffuse sphere results in slightly better scores than simply removing the geometric maps, justified by the fact that the diffuse sphere helps predict the correct dynamic range only, when the geometric information is correctly understood.

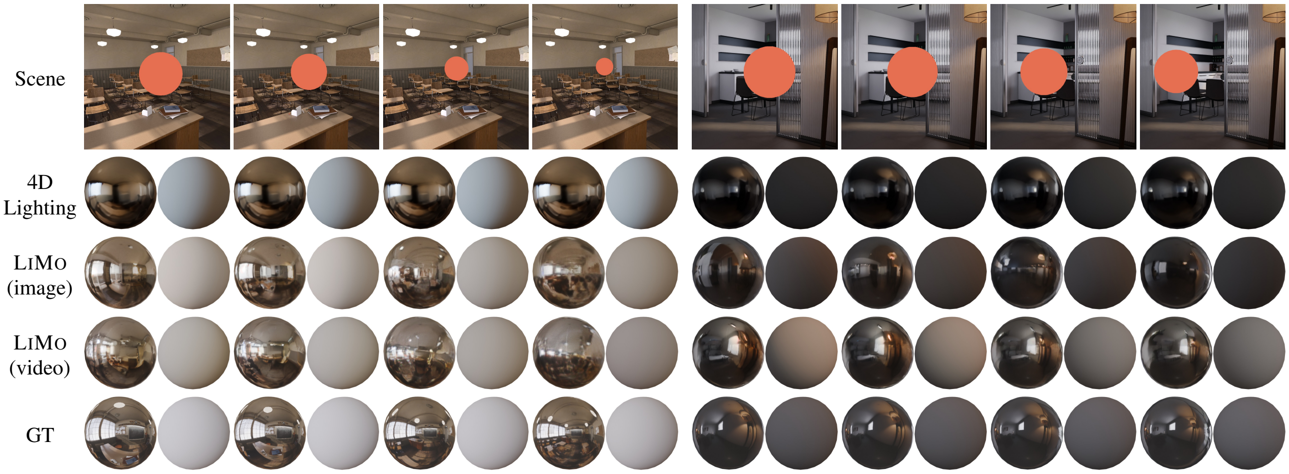

Figure 5. Example qualitative prediction results on our proposed video test set. Observe how our predictions are more detailed and more closely match the ground truth than the previous work 4D Lighting [41] as the sphere moves around the scene.

	RMSE↓				Si-RMSE↓				SSIM↑				Ang Err↓			
Method	Mirr	Diff	Gloss	Mat	Mirr	Diff	Gloss	Mat	Mirr	Diff	Gloss	Mat	Mirr	Diff	Gloss	Mat
w/o Diffuse, Geo	0.262	0.210	0.219	0.209	0.442	0.131	0.151	0.200	0.770	0.942	0.926	0.938	5.15	3.60	3.56	3.37
w/o Diffuse	0.253	0.207	0.215	0.204	0.431	0.127	0.145	0.189	0.776	0.943	0.929	0.939	4.95	3.39	3.35	3.19
w/o Geo	0.259	0.229	0.230	0.213	0.431	0.137	0.162	0.211	0.772	0.936	0.923	0.935	4.77	3.13	3.04	3.07
LiMo (full)	0.247	0.160	0.164	0.169	0.403	0.107	0.129	0.176	0.783	0.951	0.939	0.946	4.35	2.25	2.20	2.42

Table 3. Ablation of the use of the added geometric maps I_{dir} and I_{dist} for predictions (see Sec. 3.2) and diffuse sphere for HDRI optimization (see Sec. 3.1) on Infinigen with our image model. "Mirr" (mirror), "Diff" (diffuse), "Gloss" (glossy) and "Mat" (matte) refer to the different test spheres (see Sec. 4.2). Results are color coded by best , second and third best..

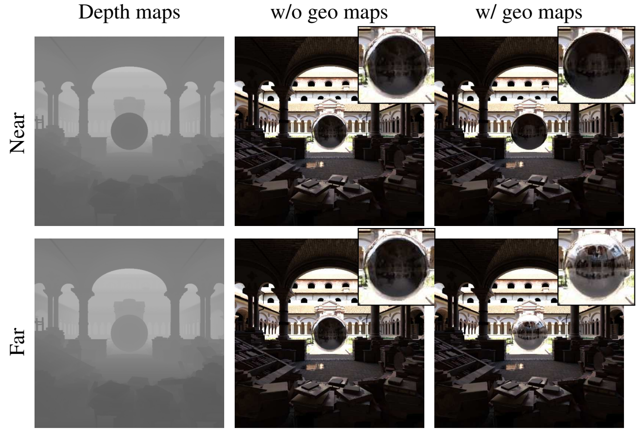

Figure 6. The effect of our proposed geometric maps $\{I_{\text{dir}}, I_{\text{dist}}\}$ on lighting predictions. We insert a sphere near (top) and farther away (bottom)—note that its screen space dimensions are the same so the network cannot use this as a cue. Observe how the addition of geometric maps (right) helps the network in reasoning about light occlusions, which are not captured otherwise (middle).

5. Conclusion

In this paper we introduced LiMo, a method for spatiotemporal scene lighting estimation. We demonstrated that combining geometrically grounded conditioning, the priors from a pre-trained diffusion model, and multiple predictions of diffuse and mirror spheres leads to state-of-the-art image and video lighting estimation. Our method provides a high level of physical accuracy and very good spatial understanding and temporal stability. Nonetheless, LiMo remains limited in several aspects. First, as rendered spheres in our dataset have a spatial extent, the HDRI optimization problem is ill-posed if a shadow is cast on the sphere or it is very close to an object as the directional lighting model is not valid anymore. Second, our method is limited to scenes and was not trained to leverage certain lighting cues such as human faces, present in many videos. Future work could account for these by adopting a truly point-based lighting representation and leveraging face datasets with known lighting.

References

[1] Jiayang Bai, Zhen He, Shan Yang, Jie Guo, Zhenyu Chen, Yan Zhang, and Yanwen Guo. Local-to-global panorama

inpainting for locale-aware indoor lighting prediction. *IEEE Trans. Vis. Comput. Graph.*, 29(11):4405–4416, 2023. 2

[2] Blender Foundation. Blender demo files. https://www.blender.org/download/demo-files/, 2025. Accessed: 2025-11-13. 5

[3] Christophe Bolduc, Yannick Hold-Geoffroy, Zhixin Shu, and Jean-François Lalonde. GaSLight: Gaussian splats for spatially-varying lighting in HDR. In *IEEE/CVF Int. Conf. Comput. Vis.*, 2025. 4

[4] Chris Careaga, S Mahdi H Miangoleh, and Yağız Aksoy. Intrinsic harmonization for illumination-aware compositing supplementary material. In *ACM SIGGRAPH Asia Conf.*, 2023. 3

[5] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2025. 2, 4

[6] Worameth Chinchuthakun, Pakkapon Phongthawee, Amit Raj, Varun Jampani, Pramook Khungurn, and Supasorn Suwajanakorn. DiffusionLight-Turbo: Accelerated light probes for free via single-pass chrome ball inpainting. In *ArXiv*, 2025. 2, 3

[7] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2025. 4

[8] Mohammad Reza Karimi Dastjerdi, Jonathan Eisenmann, Yannick Hold-Geoffroy, and Jean-François Lalonde. EverLight: Indoor-outdoor editable HDR lighting estimation. In *IEEE/CVF Int. Conf. Comput. Vis.*, 2023. 2

[9] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH Conf.*, 1998. 1, 2

[10] Paul Debevec, Chris Tchou, Andrew Gardner, Tim Hawkins, Charis Poullis, Jessi Stumpfel, Andrew Jones, Nathaniel Yun, Per Einarsson, Therese Lundgren, Marcos Fajardo, and Philippe Martinez. Estimating surface reflectance properties of a complex scene under captured natural illumination. *ACM Trans. Graph.*, 2004. 3

[11] Paul E. Debevec and Jitendra Malik. *Recovering High Dynamic Range Radiance Maps from Photographs*. Association for Computing Machinery, New York, NY, USA, 1 edition, 2023. 1

[12] Petr Dlouhý, Vilém Duha, Karolína Húserková, Monika Rygalová, Adam Krhánek, Eliška Pantůčková, Alex Hapon, Mike Radjabov, Amanpreet Bajwa, and Andreas Gajdošík. *BlenderKit: 3D assets*, 2025. 4

[13] Farshad Einabadi, Jean-Yves Guillemaut, and Adrian Hilton. Deep neural models for illumination estimation and relighting: A survey. *Comput. Graph. Forum*, 40(6):315–331, 2021. 2

[14] James A Ferwerda, Jeremy Selan, and Fabio Pellacini. Perception of lighting errors in image compositing. In *IS&T Color Imag. Conf.*, 2010. 1

[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. *ACM Trans. Graph.*, 36(6), 2017. 2

[16] Marc-Andre Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagne, and Jean-Francois Lalonde. Deep parametric indoor lighting estimation. In *IEEE/CVF Int. Conf. Comput. Vis.*, 2019. 2

[17] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-Francois Lalonde. Fast spatially-varying indoor lighting estimation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2019. 2, 5, 6, 3

[18] Diego Gomez, Julien Philip, Adrien Kaiser, and Élie Michel. Rrm: Relightable assets using radiance guided material extraction. In *Comp. Graph. Int. Conf.*, 2024. 5

[19] Zonghui Guo, Haiyong Zheng, Yufeng Jiang, Zhaorui Gu, and Bing Zheng. Intrinsic image harmonization. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2021. 3

[20] Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-François Lalonde. Deep sky modeling for single image outdoor lighting estimation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[21] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 5

[22] Jean-François Lalonde, Alexei A Efros, and Srinivasa G Narasimhan. Estimating the natural illumination conditions from a single outdoor image. *Int. J. Comput. Vis.*, 98(2): 123–145, 2012. 2

[23] 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 *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[24] Chloe LeGendre, Wan-Chun Ma, Rohit Pandey, Sean Fanello, Christoph Rhemann, Jason Dourgarian, Jay Busch, and Paul Debevec. Learning illumination from diverse portraits. In *SIGGRAPH Asia 2020 Technical Communications*, New York, NY, USA, 2020. Association for Computing Machinery. 2

[25] 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 *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2020. 2

[26] Zhengqin Li, Li Yu, Mikhail Okunev, Manmohan Chandraker, and Zhao Dong. Spatiotemporally consistent HDR indoor lighting estimation. *ACM Trans. Graph.*, 42(3):1–15, 2023. 2

[27] 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. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2025. 3

[28] Ruofan Liang, Kai He, Zan Gojcic, Igor Gilitschenski, Sanja Fidler, Nandita Vijaykumar, and Zian Wang. LuxDiT: Lighting estimation with video diffusion transformer. In *ArXiv*, 2025. 2

[29] Yiqun Mei, Mingming He, Li Ma, Julien Philip, Wenqi Xian, David M George, Xueming Yu, Gabriel Dedic, Ahmet Levent Taşel, Ning Yu, Vishal M Patel, and Paul Debevec. Lux post facto: Learning portrait performance relighting with conditional video diffusion and a hybrid dataset. *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2025. 5

[30] 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 *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2024. 2, 3, 4, 5, 6

[31] Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, Zeyu Ma, and Jia Deng. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2024. 5, 6

[32] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Comp. graph. appl.*, 21(5):34–41, 2002. 1

[33] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. *High dynamic range imaging*. Morgan Kaufman, 2005. 2

[34] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Illumination from shadows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25 (3):290–300, 2003. 2

[35] Shuran Song and Thomas Funkhouser. Neural illumination: Lighting prediction for indoor environments. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[36] Pratul P Srinivasan, Ben Mildenhall, Matthew Tancik, Jonathan T Barron, Richard Tucker, and Noah Snavely. Lighthouse: Predicting lighting volumes for spatially-coherent illumination. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2020. 2

[37] Jessi Stumpfel, Andrew Jones, Andreas Wenger, Chris Tchou, Tim Hawkins, and Paul Debevec. Direct hdr capture of the sun and sky. In *ACM SIGGRAPH Courses*, 2006. 3

[38] Minghui Tan, Jean-François Lalonde, Lavanya Sharan, Holly Rushmeier, and Carol O'Sullivan. The perception of lighting inconsistencies in composite outdoor scenes. *ACM Trans. Appl. Percept.*, 12(4), 2015. 1

[39] 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 *Eur. Conf. Comput. Vis.*, 2022. 2

[40] Zachary Teed and Jia Den. Raft: Recurrent all-pairs field transforms for optical flow. In *Eur. Conf. Comput. Vis.*, 2020. 6

[41] Mutian Tong, Rundi Wu, and Changxi Zheng. Spatiotemporally consistent indoor lighting estimation with diffusion priors. In *ACM SIGGRAPH Conf.*, 2025. 2, 4, 5, 6, 7, 8

[42] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 5

[43] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025. 4

[44] Zian Wang, Jonah Philion, Sanja Fidler, and Jan Kautz. Learning indoor inverse rendering with 3D spatially-varying lighting. In *IEEE/CVF Int. Conf. Comput. Vis.*, 2021. 2

[45] Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. RGB↔X: Image decomposition and synthesis using material- and lighting-aware diffusion models. In *ACM SIGGRAPH Conf.*, 2024. 3

[46] Fangneng Zhan, Changgong Zhang, Yingchen Yu, Yuan Chang, Shijian Lu, Feiying Ma, and Xuansong Xie. EMLight: Lighting estimation via spherical distribution approximation. In *Assoc. Adv. of Art. Int.*, 2021. 2

[47] Jinsong Zhang, Kalyan Sunkavalli, Yannick Hold-Geoffroy, Sunil Hadap, Jonathan Eisenman, and Jean-François Lalonde. All-weather deep outdoor lighting estimation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[48] Zitian Zhang, Frédéric Fortier-Chouinard, Mathieu Garon, Anand Bhattad, and Jean-François Lalonde. ZeroComp: Zero-shot object compositing from image intrinsics via diffusion. In *IEEE/CVF Winter Conf. App. Comput. Vis.*, 2025. 3

[49] Rui Zhu, Zhengqin Li, Janarbek Matai, Fatih Porikli, and Manmohan Chandraker. Irisformer: Dense vision transformers for single-image inverse rendering in indoor scenes. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2022. 2

[50] Yongjie Zhu, Yinda Zhang, Si Li, and Boxin Shi. Spatially-varying outdoor lighting estimation from intrinsics. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2021. 2

Lighting in Motion: Spatiotemporal HDR Lighting Estimation

Supplementary Material

6. Data generation details

We use Blender, paired with BlenderKit assets to procedurally generate indoor and outdoor renders. For indoor scenes, we use the full indoor scenes provided by BlenderKit, generating more cameras based on the original ones. Since scenes are not always completely modeled, we leverage existing camera assuming they point toward points of interest. We randomly sample a direction from the original camera frustrum to obtain a target lookat point using ray-casting. Then we sample 3D points in the scene bounding box and check the visibility of the selected lookat point from them. If the point is visible from the sampled position, we consider the location to be a valid camera location and create a new camera pointed at the lookat.

For outdoor scenes, we select a central model from the building, vehicle or nature categories of BlenderKit. We add a ground plane, a random ground material, and add surrounding buildings, objects and vegetation using particle systems. We use HDRis from Polyhaven for lighting. We then derive cameras pointed at the central object from which it is visible. In total we use around 500 indoor scenes, reusing them for different motion for a total of 4400 scenes, and generate 1200 outdoor scenes. For each scene we render 4 viewpoints.

7. HDRI map optimization details

The predicted images from the network $\hat{I}$ are cropped around the inpainted spheres. The same is done with the sphere mask, normals and position maps. The equirectangular HDRI is a Laplacian pyramid at a fixed resolution of 512x256 with 8 levels. We employ circular padding to leverage the cyclic nature of equirectangular maps. For faster convergence and better conditioning, the HDRI is defined in $\log_2$ space. We optimize with Adam using a learning rate of $5\mathrm{e}{-3}$ for 1000 iterations per frame, for a total of 21 000 steps.

At every step, we randomly select a frame t, sphere type m (mirror or diffuse) and EV e from the predictions. The Laplacian pyramid is recomposed and is transformed back to linear space to obtain the HDRI map L_t. Then, the renderer $\mathcal{R}$ is used to produce the image of corresponding sphere (mirror or diffuse). This rendered image is exposed according to the randomly selected EV and converted to sRGB color space to match the network's prediction's colors:

$$\hat{I}_t = sRGB(2^e \mathcal{R}((L_t, m))). \tag{7}$$

The loss function to optimize the HDRI representation is defined as:

$$\ell = M_{\text{sat}}(\ell_2(\hat{I}_t, I_t) + \frac{\lambda}{2}(\ell_1(\hat{I}_t, \hat{I}_{t-1}) + \ell_1(\hat{I}_t, \hat{I}_{t+1}))), \tag{8}$$

with $\lambda = 0.1$ in all our experiments. The ℓ_2 loss enforces the rendered image to closely match the predicted image, and the two following ℓ_1 losses insure that the rendered image be similar to the neighboring frames, allowing for temporal smoothing. To prevent the saturated part of the image from lowering the overall intensity of the optimized HDRI, we define a saturation mask

$$M_{\text{sat}} = \begin{cases} 0, & \hat{I}_t > \tau \text{ and } I_t > \tau, \\ 1, & \text{otherwise.} \end{cases} \tag{9}$$

The renderer $\mathcal{R}$ is a two modes differentiable Monte Carlo renderer for perfect mirror and perfect diffuse materials. The perfect reflection is implementing the reflection equation

$$\mathbf{v}_i - 2(\mathbf{v}_i \cdot \mathbf{n}_i)\,\mathbf{n}_i, \tag{10}$$

with $\mathbf{v}_i$ computed from the sphere's position map.

For the diffuse rendering, we first compute the luminance of the HDRI to use as importance weight:

$$L = 0.2126R + 0.7152G + 0.0722B \tag{11}$$

The importance map for each pixel of the HDRI map is computed as a multi-importance weighting of cosine and luminance:

$$w_i = (n_i \cdot r_i)\, L_i\, sin(r_i), \tag{12}$$

where r_i is the ray direction corresponding to pixel i of the HDRI map. It is then normalized:

$$\mathbf{w} = \frac{w}{\sum_i w_i}. \tag{13}$$

The corresponding probability distribution function is computed by dividing the normalized importance map by the solid angle of the equirectangular map

$$PDF = \frac{\mathbf{w}}{\partial \omega}. \tag{14}$$

Samples s are drawn from the importance map $\mathbf{w}$ and the final rendered colors R_i is

$$R_i = \frac{1}{S} \sum_{s \in S} \frac{L_s(n_i \cdot r_i)}{PDF}. \tag{15}$$

We use 64 samples with sub-pixel sampling in all our experiments.

		RMSE$_\downarrow$		SI-RMSE$_\downarrow$		SSIM$_\uparrow$		Ang. Err.$_\downarrow$		T-LPIPS$_\downarrow$		T-LPIPS-Diff$_\downarrow$		Warped Err$_\downarrow$	
Dataset	Method	Gloss	Mat	Gloss	Mat	Gloss	Mat	Gloss	Mat	Gloss	Mat	Gloss	Mat	Gloss	Mat
Dynamic object	4D Lighting	0.30	0.33	0.21	0.34	0.89	0.88	4.6	4.9	0.0009	0.0006	0.0064	0.0016	0.0125	0.0099
	LiMo (image)	0.15	0.16	0.12	0.17	0.96	0.97	1.4	1.7	0.0224	0.0138	0.0166	0.0117	0.0489	0.0575
	LiMo (video)	0.18	0.21	0.13	0.21	0.96	0.96	2.9	3.0	0.0025	0.0020	0.0053	0.0017	0.0200	0.0156
Dynamic camera	4D Lighting	0.38	0.37	0.18	0.31	0.89	0.88	3.8	4.3	0.0011	0.0010	0.0047	0.0007	0.0163	0.0142
	LiMo (image)	0.16	0.17	0.12	0.18	0.96	0.97	1.4	1.8	0.0179	0.0114	0.0125	0.0102	0.0499	0.0541
	LiMo (video)	0.21	0.23	0.13	0.21	0.96	0.96	2.4	3.0	0.0019	0.0015	0.0042	0.0012	0.0178	0.0150
Dynamic lighting	4D Lighting	0.34	0.35	0.70	0.80	0.86	0.85	10.1	10.1	0.0008	0.0007	0.0011	0.0005	0.0095	0.0096
	LiMo (image)	0.17	0.18	0.13	0.19	0.95	0.96	1.8	2.2	0.0032	0.0022	0.0018	0.0015	0.0203	0.0217
	LiMo (video)	0.22	0.25	0.15	0.24	0.94	0.95	2.8	3.2	0.0009	0.0008	0.0010	0.0005	0.0077	0.0077
Combination	4D Lighting	0.32	0.33	0.20	0.33	0.89	0.88	4.0	4.3	0.0017	0.0013	0.0103	0.0018	0.0217	0.0167
	LiMo (image)	0.16	0.18	0.12	0.19	0.96	0.97	1.8	2.2	0.0249	0.0170	0.0137	0.0140	0.0615	0.0723
	LiMo (video)	0.23	0.24	0.16	0.24	0.94	0.94	2.7	3.0	0.0048	0.0042	0.0072	0.0033	0.0337	0.0289

Table 4. Quantitative evaluation of lighting estimation on dynamic scenes for "Gloss" (glossy) and "Mat" (matte) spheres in complement to Tab. 2. We compare LiMo with "4D Lighting" [41]. Results are color coded by best, second best.

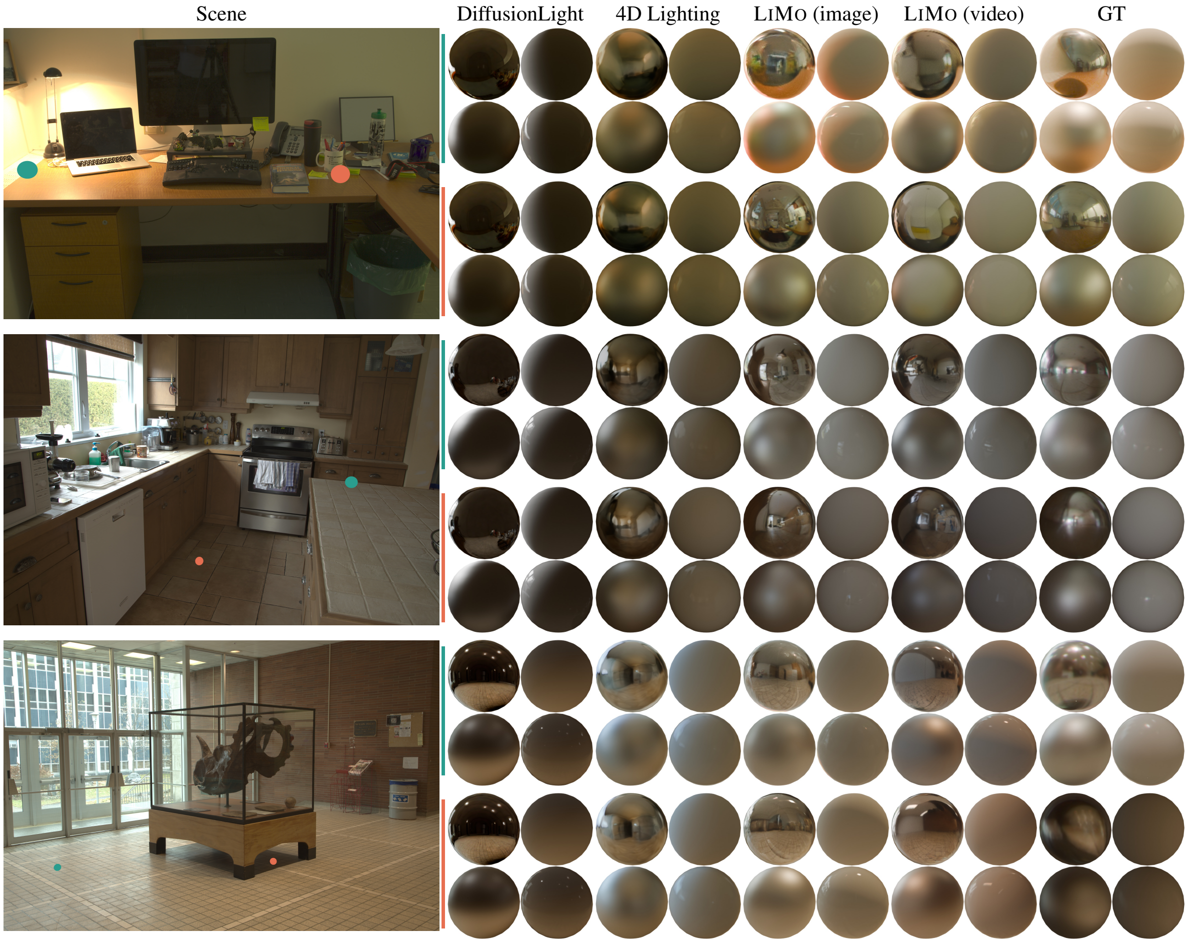

Figure 7. Additional sample predictions from the Laval Indoor Spatially Varying test set [17] for, from left to right: DiffusionLight [30], 4D Lighting [41], and the image and video versions of the proposed LiMo. We visualize predictions by rendering the same four test spheres used for the quantitative metrics (see Tab. 1): mirror (top left), diffuse (top right), matte (bottom left) and glossy (bottom right).

8. Additional results

In complement to Tab. 2, Tab. 4 reports metrics on our sequences test dataset for glossy and matte spheres.

Sample predictions from The Laval Indoor Spatially Varying HDR dataset [17] are presented in Fig. 7.

More in-the-wild results are presented in Fig. 8. We make use of the predicted pointcloud from the FOV and depthmap as shadow catcher when inserting objects in the scene.

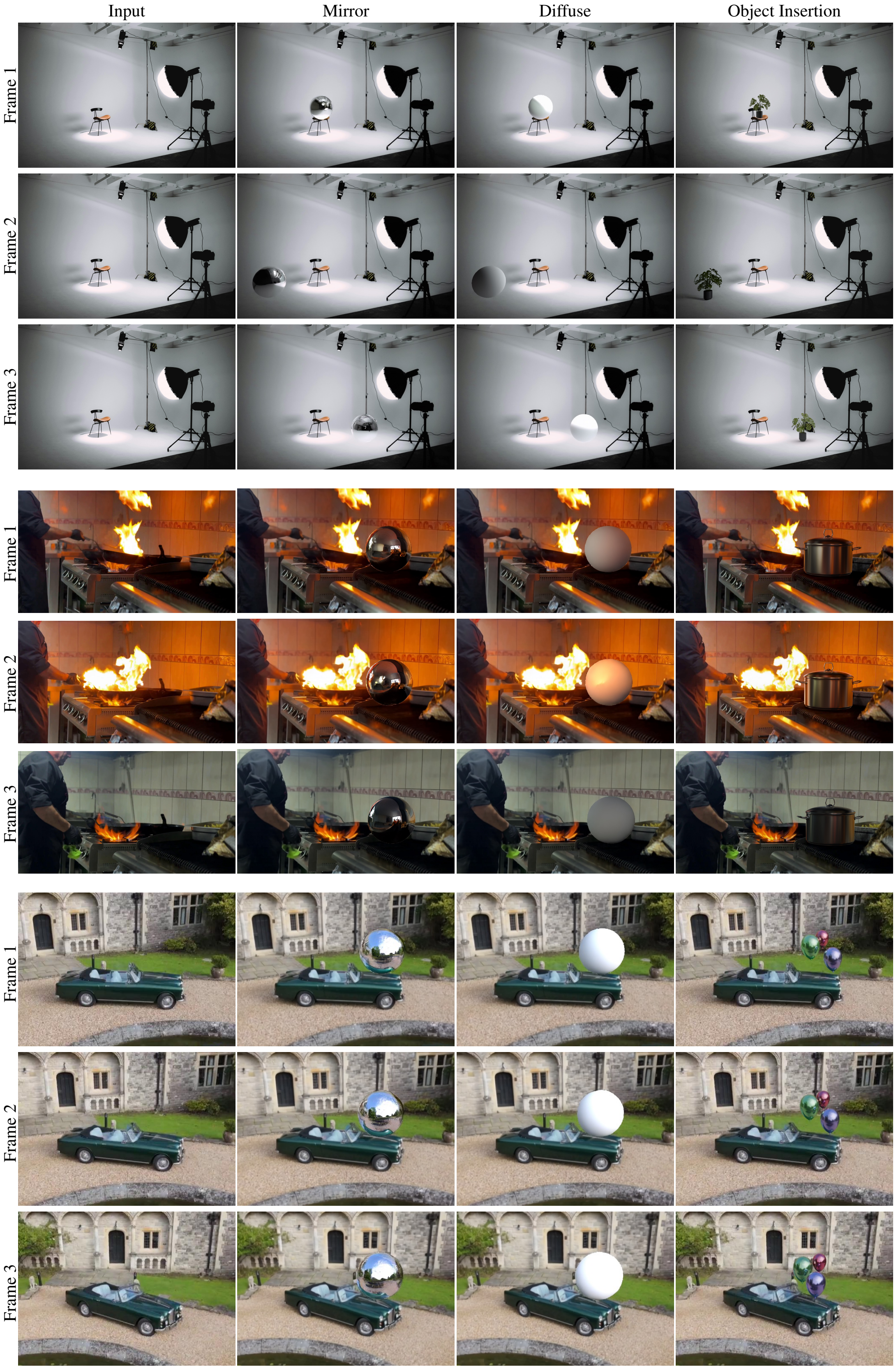

Figure 8. Additional examples of our method on in-the-wild images and videos, with from left to right: the input frame, the predicted mirror sphere at EV0, the predicted diffuse sphere at EV0 and the inserted object. The predicted pointcloud is used as shadow catcher.