A Generalizable Light Transport 3D Embedding for Global Illumination

BING XU, University of California, San Diego, USA and NVIDIA, USA
MUKUND VARMA T, University of California, San Diego, USA
CHENG WANG, University of California, San Diego, USA
TZUMAO LI, University of California, San Diego, USA
LIFAN WU, NVIDIA, USA
BARTLOMIEJ WRONSKI, NVIDIA, USA
RAVI RAMAMOORTHI, University of California, San Diego, USA
MARCO SALVI, NVIDIA, USA

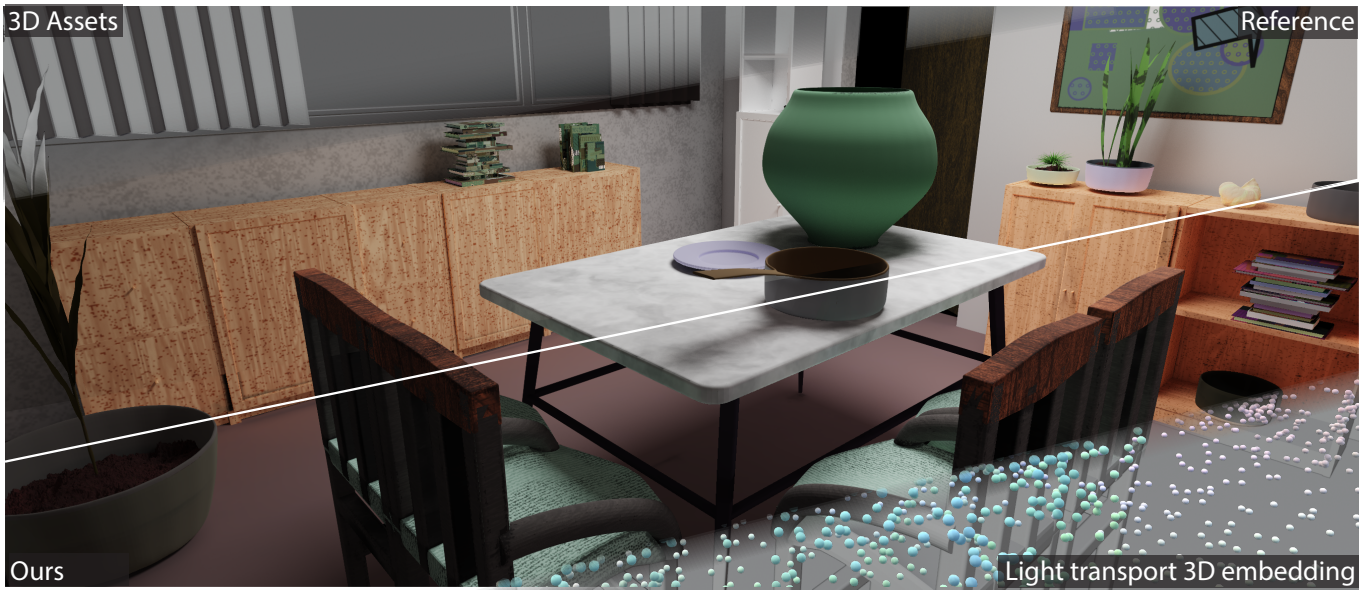


Fig. 1. Our transformer-based, generalizable light-transport model (LTM) takes as input the 3D scene assets (top-left slice)—including geometry, materials, and lighting represented as a point cloud—and encodes light transport into a 3D embedding (bottom-right slice), defined by latent codes anchored at sampled scene points. The model is view- and resolution-independent. The middle two slices show the path-traced reference and ours with predicted global illumination result at 2688 × 1152 resolution for an unseen test scene.

Global illumination (GI) is essential for realistic rendering but remains computationally expensive due to the complexity of simulating indirect light transport. Recent neural methods for GI have focused on per-scene optimization with extensions to handle variations such as dynamic camera views or geometries. Meanwhile, cross-scene generalization efforts have largely remained confined in 2D screen space—such as neural denoising or G-buffer–based GI prediction—which often suffer from view inconsistency and limited spatial awareness.

In this paper, we learn a generalizable 3D light transport embedding that directly approximates global illumination from 3D scene configurations, without utilizing rasterized or path-traced illumination cues. We represent each scene as a point cloud with features and encode its geometric and material interactions into neural primitives by simulating global point-to-point long-range interactions with a scalable transformer. At render time, each query point retrieves only local primitives via nearest-neighbor search and adaptively aggregates their latent features through cross-attention to predict the desired rendering quantity.

We demonstrate results for estimating diffuse global illumination on a large indoor-scene dataset, generalizing across varying floor plans, geometry, textures, and local area lights. We train the embedding on irradiance prediction and demonstrate that the model can be quickly re-targeted to new rendering tasks with limited fine-tuning. We present preliminary results on spatial-directional incoming radiance field estimation to handle glossy materials, and use the normalized radiance field to jump-start path guiding for unbiased path tracing. These applications point toward a new pathway for integrating learned priors into rendering pipelines, demonstrating the capability of predicting complex global illumination in general scenes without explicit ray traced illumination cues.

Authors' Contact Information: Bing Xu, b4xu@ucsd.edu, University of California, San Diego, USA and NVIDIA, USA; Mukund Varma T, tmukund@ucsd.edu, University of California, San Diego, USA; Cheng Wang, chengwang@ucsd.edu, University of California, San Diego, USA; Tzumao Li, tzli@ucsd.edu, University of California, San Diego, USA; Lifan Wu, lifanw@nvidia.com, NVIDIA, USA; Bartlomiej Wronski, elirian@gmail.com, NVIDIA, USA; Ravi Ramamoorthi, ravir@cs.ucsd.edu, University of California, San Diego, USA; Marco Salvi, msalvi@nvidia.com, NVIDIA, USA.

1 Introduction

A long-term goal of computer graphics has been to efficiently simulate light transport while preserving the fidelity of the complex and exquisite interaction of light and matter. Over the years, researchers have derived mathematical equations and computational models to describe various underlying physical phenomena, including light emission, reflections, global light transport, caustics, and color bleeding [Christensen et al. 2016]. Recent years have witnessed an alternative paradigm, showcasing the power of data priors to approximately extract or predict the underlying physical processes, as shown in recent generative models [OpenAI 2024]. With the growing availability of 3D data including generated assets [Siddiqui et al. 2024], we seek to answer a pivotal question: whether data priors alone are sufficient to train a generalizable light-transport model for global illumination—one that can (partially) replace 3D world space light transport simulation—merely based on scene configurations.

Monte Carlo rendering remains the workhorse of photo-realistic imagery. Recent works on neural global illumination [Diolatzis et al. 2022; Granskog et al. 2020; Rainer et al. 2022; Ren et al. 2013; Zheng et al. 2023] approximates light transport using feed-forward networks, enabling far more efficient reuse of lighting computations. However, these methods often overfit to individual scenes, limiting scalability and generalization. Two lines of work that attempt cross-scene generalization operate on the 2D domain, using neural denoisers [Áfra 2024], or inferring final shading from G-buffer features [Nalbach et al. 2017]. However, these methods are prone to view inconsistency and struggle to capture global illumination from off-screen geometry.

In this paper, we tackle the challenging problem of estimating global illumination in a generalizable way across diverse scene layouts, geometries, textures, and lighting, while maintaining multi-view consistency.

Our approach draws inspiration from the success of transformers [Vaswani et al. 2017] to capture long-range interactions and explore their practicality in light transport simulation where the lights, geometries and materials exhibit complex relations. Specifically, given input geometry, material properties, and area light sources, our method first samples the scene into a point cloud. We then apply our point-based scalable transformer to encode the scene into *light transport embeddings*, latent codes anchored at scene points. Given a query point (ray intersection point), our decoder then collects the nearest latent codes, and aggregates them adaptively to estimate the irradiance or incoming radiance fields. The whole process is end-to-end differentiable and can be supervised using ground truth rendering quantities at sampled query points in the 3D scene. We train the base embedding on diffuse global illumination and show that while freezing the encoding, we can simply fine-tune the decoding stage with limited training iterations and still manage to faithfully recover glossy reflections. In addition, we demonstrate that the predicted radiance field can be used as an initial importance distribution to bootstrap path guiding before sufficient samples are gathered in the usual per-scene optimization setting. This highlights its potential to accelerate convergence toward unbiased solutions.

Concretely, we make the following contributions:

(1) We learn a generalizable light transport model in a view-independent manner, directly from 3D configurations without rasterized or path traced illumination cues, achieving high fidelity to path-traced references on unseen scenes.
(2) We curate and will release a large-scale indoor dataset (~14k scenes) with diverse floorplans, geometries, and textures, serving as a benchmark for learning light transport.
(3) We propose several design choices, including a point-based neural primitive as an intermediate representation, a scalable transformer-based light transport encoder to encode long-range interactions between scene points for global illumination effects, task-specific local query decoders aided by cross-attention, and a view-independent training scheme. (Sec. 4.1,4.2,4.3)
(4) We demonstrate the versatility of our framework by reusing a shared pretrained encoder and introducing new task-specific decoders. The same model estimates diffuse global illumination that generalizes across complex indoor scenes. With a newly switched decoder and limited fine-tuning, we present preliminary results on glossy materials by predicting spatial-directional incident radiance fields, and further explore extensions to bootstrap path guiding for unbiased GI. (Sec. 6, 7)

2 Related work

We focus our discussions on data-driven methods, especially the ones that make use of neural networks, that attempt to approximate light transport, or operate on 3D data.

Precomputation-based rendering. Precomputation-based methods accelerate global illumination by shifting transport computation offline. Classical precomputed radiance transfer (PRT) [Ramamoorthi et al. 2009; Sloan et al. 2002a] compresses the light transport matrix and models scene response to dynamic lighting by projection to linear basis. Direct-to-indirect transfer methods [Hašan et al. 2006] precompute mappings from direct to indirect illumination, enabling real-time GI computation at runtime. Recent neural PRT methods [Raghavan et al. 2023; Rainer et al. 2022] fit scene-specific networks to approximate transport under varying lighting. This type of precomputation usually needs to be done per scene, while our method learns a generalizable light transport embedding that can be reused across various scenes.

Overfitting to a single scene. The advent of deep learning has spurred numerous approaches that replace or augment parts of the classical rendering pipeline with neural networks. Early work by Ren et al. [2013] fits a radiance regression function to a single scene for fast indirect illumination. Subsequent research [Diolatzis et al. 2022; Eslami et al. 2018; Gao et al. 2023; Granskog et al. 2020; Rainer et al. 2022; Zheng et al. 2024, 2023] focuses on improving the expressiveness of neural models in terms of handling a subsets of the varying factors: dynamic scenes with moving objects, varying materials, changing viewpoints, or dynamic lighting. Most of these methods require per-scene training. In contrast, our approach focuses on scene-level generalization and scalability, using only raw 3D scene configurations without external shading cues.

Generalizable neural techniques for rendering. Neural denoising and deep shading are two major approaches that aim for scene-level generalization. Denoising methods [Bako et al. 2017; Chaitanya et al. 2017; Gharbi et al. 2019; Vogels et al. 2018; Xu et al. 2019] operate purely as a post-processing step, leaving the underlying light transport simulation unchanged. Deep shading method [Nalbach et al. 2017] infers final shading from G-buffer features in screen space. Xin et al. [2022] extend this by using a bilateral CNN to predict single-bounce indirect illumination from direct lighting information.

Despite their distinct mechanisms, both types of approaches share a common limitation: their reliance on screen-space input can lead to view inconsistencies and difficulties in handling occluded or off-screen objects that nonetheless contribute to global illumination. Our method addresses these issues by operating on full 3D scene configurations and learning a generalizable world-space light transport embedding, enabling coherent illumination across diverse scenes and viewpoints. While Hermosilla et al. [2019] aim for a similar goal, their scope is limited to single-object scenes and convolutional network architectures. Our new transformer-based encoder is designed to scale to complex, multi-object scenes and can handle dynamic local lighting, offering broader applicability.

Concurrent work by Zeng et al. [Zeng et al. 2025] explores a related idea of learning light transport using transformer architectures. Their method employs two global transformers applied to mesh vertices, effectively treating vertex positions as sampled points on the geometry. While their approach is limited to relatively small meshes (up to 5k vertices) due to the quadratic cost of vanilla transformers, our architecture is designed to scale to much denser point representations. Another key distinction lies in the training supervision: their model is trained on 2D rendered images, making it view- and resolution-dependent, whereas our approach predicts incident radiance fields directly in 3D intead of outgoing radiance. The two approaches therefore complement each other: their work emphasizes glossy material rendering in controlled Cornell-box scenes, while we target diverse, complex indoor environments with varying spatial textures and flexible scene scales, enabling unrestricted walkarounds with view and resolution independence. Looking ahead, a shared challenge for both methods is to further scale up with rendering parameters to capture the full spectrum of light transport effects.

Learning with irregular geometric data. Learning with irregular 3D data such as point clouds poses unique challenges. Models must be invariant to input permutation and robust to variations in point density and topology. Early seminal works like PointNet [Qi et al. 2017a] and PointNet++ [Qi et al. 2017b] apply shared MLPs and hierarchical feature grouping to unordered point sets. Subsequently, graph-based models [Wang et al. 2019] and continuous convolutional operators [Li et al. 2019; Thomas et al. 2019] are developed to capture local geometry through learned neighborhood functions. Attention-driven architectures such as Point Transformer [Zhao et al. 2021] further enhance the integration of global context. Our work uses a transformer-based architecture built upon PointTransformerV3 [Wu et al. 2024]. This choice prioritizes scalability and architectural simplicity, enabling efficient learning of complex relationships within 3D point clouds for global illumination.

3 Motivation and Overview

3.1 Light Transport Preliminaries

A central challenge in photorealistic rendering is the accurate simulation of physically based light transport, mathematically formulated by the rendering equation [Kajiya 1986]:

$$L(\mathbf{x}, \boldsymbol{\omega}_o) = L_e(\mathbf{x}, \boldsymbol{\omega}_o) + \underbrace{\int_{\mathcal{H}^2} f_r(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L_i(\mathbf{x}, \boldsymbol{\omega}_i)(\mathbf{n}_\mathbf{x} \cdot \boldsymbol{\omega}_i)\, \mathrm{d}\boldsymbol{\omega}_i}_{=: L_r(\mathbf{x}, \boldsymbol{\omega}_o)}, \tag{1}$$

where L is the total outgoing radiance, L_e denotes the emitted radiance, $\mathcal{H}^2$ is the hemispherical integration domain, f_r represents the spatially-varying bidirectional reflectance distribution function (SVBRDF), L_i is the incident radiance, and $\mathbf{n}_\mathbf{x}$ denotes the normal at $\mathbf{x}$. The reflected radiance $L_r(\mathbf{x}, \boldsymbol{\omega}_o)$ can be further decomposed as [Nayar et al. 2006; Sloan et al. 2002b]:

$$L_r(\mathbf{x}, \boldsymbol{\omega}_o) = L^0(\mathbf{x}, \boldsymbol{\omega}_o) + L^+(\mathbf{x}, \boldsymbol{\omega}_o), \tag{2}$$

where L^0 is the direct illumination component that accounts for radiance directly from light sources after a single scattering event at $\mathbf{x}$, and L^+ is the global illumination component capturing light that has undergone multiple scattering events throughout the scene.

Direct illumination is often easier to compute, since it involves sampling only a single bounce to light sources explicitly. Consequently, the main challenge in many rendering scenarios lies in efficiently and accurately computing global illumination, which involves a recursive integral and captures complex visual effects such as indirect lighting, soft shadows, and inter-reflections. Our work primarily focuses on the approximation of indirect illumination–termed global illumination $L^+(\mathbf{x}, \boldsymbol{\omega}_o)$.

3.2 An Analogy Bridging Light Transport and Attention

We motivate our approach by seeking similar patterns and an analogy between light transport and the attention mechanism [Vaswani et al. 2017]. Note that this analogy is intended to provide conceptual insight and guide our model design, rather than to define a mathematically exact isomorphism. Fig. 2 offers a visual intuition for some of these parallels, illustrating how attention patterns extracted from our method can highlight relevant scene regions contributing to global illumination.

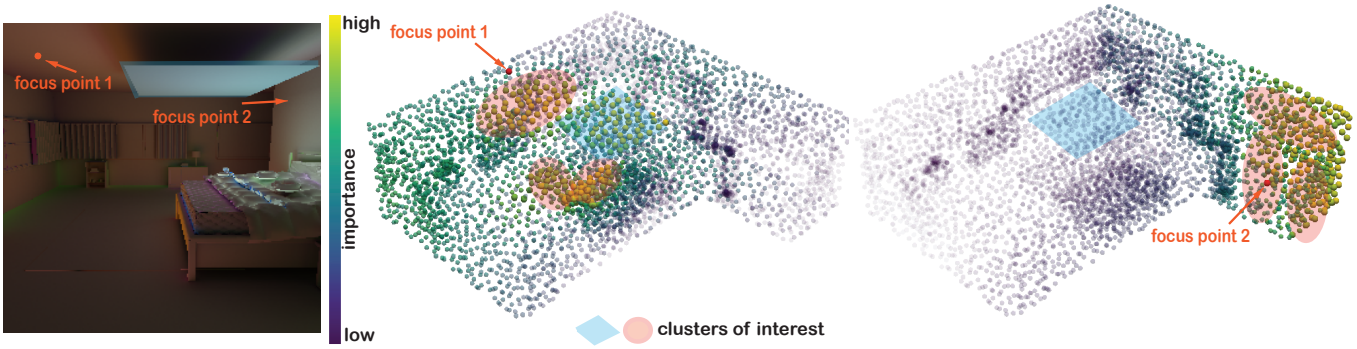
The operator formulation of light transport [Arvo et al. 1994; Veach 1998] reads

$$\mathbf{l}_{\text{out}} = (I - T)^{-1}\mathbf{l}_e = \mathbf{l}_e + T\mathbf{l}_e + T^2\mathbf{l}_e + \cdots, \tag{3}$$

where $\mathbf{l}_{\text{out}}$ is the vector of outgoing radiance, $\mathbf{l}_e$ is emitted radiance, I is an identity matrix, and T is the light transport matrix. Hence, light transport can be seen as a series of matrix-vector multiplications, accumulated on top of each other.

On the other hand, the mechanism of transformer self-attention is expressed as

$$\text{Attention}(Q, K, V) = AV, \text{ where } A_{ij} = \frac{\exp\left(q_i \cdot k_j / \sqrt{d}\right)}{\sum_{j'} \exp\left(q_i \cdot k_{j'} / \sqrt{d}\right)}. \tag{4}$$



(a) Irradiance rendering from one view; see the light source above the bed, highlighted in blue.

(b) Attention heatmap for focus point 1 located on the ceiling next to the light.

(c) Focus point 2 on the wall of the hallway; occluded from the light source.

Fig. 2. Visualization of attention for a bedroom floorplan with a hallway. We show the attention heatmaps for two different focus scene points. The points are shaded using importance as the alpha value whereas the top 200 points with the highest attention scores are larger. The attention is extracted from our light transport encoder. The blue shaded square denotes the light source. In (b), we see four groups of scene points having the largest impact on the focus point (highlighted in blue square and orange clusters), which likely corresponds to 4 bounces of ray tracing. In (c), the focus point is on the wall of the hallway where the light source is occluded, so no points on the light source contribute much. Points with high attention scores are mostly located on different walls in the hallway, also forming three groups (highlighted in orange clusters). This serves as a visual hint for our analogy between the light transport operator and the attention mechanism in section 3.2.

Here, Q, K, V are matrices derived from input embeddings. The act of multiplying the light transport matrix T by a radiance vector resembles the attention mechanism, where each position-direction pair would need to query over all other positions to accumulate global illumination contributions. Likewise, in an attention layer, each query vector computes its attention scores by interacting with all key vectors, enabling the model to capture long-range, sequence-wide (or scene-wide in the context of rendering) dependencies directly. This inherent capability of modeling global interactions is the foundation of both light transport simulation and the expressive power of transformers. To justify this analogy, Figure 2 visualizes how the attention mechanism identifies key surface points that are important to a particular focus point during multi-bounce light transport simulation. This analogy offers insight into why transformers may approximate light transport, and suggests some further extensions —such as sparsity and importance sampling—could correspond to sparse attention to focus on the important tokens, cutting down on all-to-all weighting while still capturing global context.

3.3 Overview

We introduce a generalizable 3D embedding for light transport simulation and a novel learning framework that predicts global illumination directly from 3D configurations without rasterized or path-traced illumination cues. Our pipeline is illustrated in Figure 3, and the rest of the paper is organized as follows.

Formally given a 3D scene, we derive a point-based representation containing per-point attributes (illustrated in section 4.1). Next, we encode these scene points into a high-dimensional neural representation ($\mathbf{F}_i$) that captures the light transport function of the given scene. Motivated by the recent success of transformer-based architectures and our observations in Sec. 3, we leverage a transformer-based encoder (dubbed *Light Transport Encoder*) to capture complex relations between the scene points and implicitly encode the light transport operator (section 4.2). To compute rendering quantities (e.g. (ir)radiance) at a given query point in the 3D scene (ray intersection point or explicitly sampled from the scene geometrys), we accumulate information from its neighboring scene point embeddings. We observe that naive aggregation leads to sub-optimal results and propose an attention-based decoder (dubbed *Local Query Decoder*, section 4.3).

Our method operates on an explicit 3D input, and maintains such a 3D internal representation throughout the entire network, ensuring multi-view consistency without any additional regularization. Moreover, this also enables scalability to any resolution inputs at render time. In section 4, we provide more details on the individual components of this pipeline.

In Section 5, we present our dataset, training scheme, and integration into a real-time rendering system at inference time. We then demonstrate several rendering applications using our learned light transport embedding.

In Section 6, we predict irradiance for diffuse global illumination based on our embedding. Our 3D neural embedding bears conceptual resemblance to irradiance caching [Ward and Heckbert 1992] where they deposit irradiance at discrete spatial positions. Yet, irradiance caching is challenged by optimal probe placement and interpolation strategies, often leading to well-known light leak artifacts (inherent to discretizations). Instead of storing scalar irradiance at discrete cache probes for static geometry, we deposit learned neural primitives and more robustly aggregate neighborhood information through the attention-based local decoders. Since irradiance has a relatively smooth spatial variation on diffuse surfaces, we choose this task to demonstrate our model's capability of cross-scene generalization. In Section 7, we present preliminary results on glossy

GI, showing that our model can be easily re-purposed to new tasks with limited fine-tuning.

Echoing the progression from irradiance caching to radiance caching [Jarosz et al. 2008], we re-target our pretrained Light Transport Encoder with another dedicated decoder that incorporates world-space directions to predict spatial-directional incoming radiance fields, thereby supporting glossy materials (section 7). Unlike spherical harmonic bases, which require extremely high orders to capture fine-grained angular detail, our learned neural basis can represent substantially higher-frequency variation more efficiently, producing more accurate glossy reflections (see the comparison in fig. 12). Building on the predicted incoming radiance field, we also show that it can serve as an initial importance sampling distribution to bootstrap path guiding, before enough samples have been gathered in a regular per-scene optimization setting. We provide preliminary results that demonstrate this capability and its potential to accelerate convergence (section 7).

4 Light transport 3D embedding

We present our design choices and describe the main components of our model.

4.1 Points as the intermediate scene representation

Given 3D assets of a scene, with information about their geometry, material properties and light sources, we can represent them in several input formats. Prior works that propose generalizable neural shaders or denoisers leverage 2D image-aligned formats that enable the reuse of existing performant 2D vision models [Nalbach et al. 2017; Áfra 2024]. However, this can inherently lead to view-inconsistency. Moreover, these models are limited to a fixed image resolution, in contrast to directly learning the light transport operator on 3D.

We adopt point clouds as our intermediate representation (IR), offering several advantages and flexibility: 1) scalability: point density can be adjusted based on importance; 2) generality: they are decoupled from the original geometry, and suitable for other sources such as scanned data; and 3) simplicity: they are easier for neural networks to handle than meshes, with no connectivity.

Light sources. Most prior works in neural global illumination have focused on distant illumination (namely, environment maps). However, this approximation does not capture near field lighting effects. We instead use local area lights to produce rich, spatially varying illumination in complex indoor scenes. By representing light sources as point clouds, we unify the treatment of emissive and non-emissive geometries. This also makes our method agnostic to the type and number of lights, allowing easy extension to sources like point lights or light strips.

Scene Points. As a pre-processing step (shown in the left part of Figure 3), we convert each scene into the intermediate representation by sampling all the object geometries. Each point contains information about the position, normals, material properties like albedo and emissivity. For each point i, this can be formally represented as $(\mathbf{p}_i, \mathbf{n}_i, \mathbf{c}_i, \mathbf{e}_i)$ and considering M ($\approx 20K$) total points in the scene, our input scene representation for all points can be denoted as $\{(\mathbf{p}_i, \mathbf{n}_i, \mathbf{c}_i, \mathbf{e}_i)\}^M$. We term these points to be *Scene Points*. Some previous work that approximates light transport also takes point cloud as input [Hašan et al. 2006; Hermosilla et al. 2019]. More discussion on point sampling of the scene can be found in section 8.

Query Points. At render time, *Query Points* are the ray-intersection points where the shading computations are performed. And during training, here are where the loss is evaluated for backpropagation. Similar to *Scene Points*, each query point has associated per-point features, denoted as $\{(\mathbf{p}_j, \mathbf{n}_j, \mathbf{c}_j)\}$. Query points have more flexibility in the sampling strategy. During training, we can collect all the intersection points from a path tracer which will be view dependent. Alternatively, they can be uniformly sampled from the scene surfaces to ensure unbiased[1] surface coverage. We chose the second approach to decouple from views and support free camera movement. N ($\approx$ 2 Million) query points are sampled per scene for training. More ablation is shown in section 8.

4.2 Light Transport Embedding

We learn to approximate global illumination effects directly from 3D scene inputs—geometries, materials, light sources—without relying on any rasterized or path-traced illumination cues. Global illumination requires simulating the full complexity of light transport, including multiple bounces against various surfaces and interactions with various material properties (BSDFs), before reaching the camera or sensor. Our observations from Sec. 3 motivate us to leverage a transformer-based encoder to derive per-point neural primitives that implicitly model the light transport operator.

Given that we require a large number of points to represent an entire scene with sufficient density ($M \approx 20$K), naively borrowing the original transformer architecture can lead to significant prohibitive memory requirements, since self-attention scales quadratically and requires storing massive $M \times M$ attention maps. Rather, we leverage state-of-the-art point cloud encoder PointTransformerV3 [Wu et al. 2024] for deriving a latent representation for each point. Our experiments indicate that naïvely using all input features as input and feeding them into the transformer yields suboptimal results. We attribute this to the attention operation becoming increasingly sparse with a large number of input tokens, leading to difficulties in capturing complex interactions between the scene points. Yet, aggressively reducing the number of input points can lead to information loss, as a large indoor scene cannot be adequately represented with a limited number of points. Therefore, we derive an embedding from a sparser set of points, while also aggregating features from their immediate neighbors. This reduces attention operation's "learning overhead" in modeling interactions over a larger number of input points, while still preserving information from all the points that originally represented the scene. Formally, we write the Nearest Neighbor Embedding (see Fig. 4(a)) operation as follows:

[1] Unbiased in terms of views.

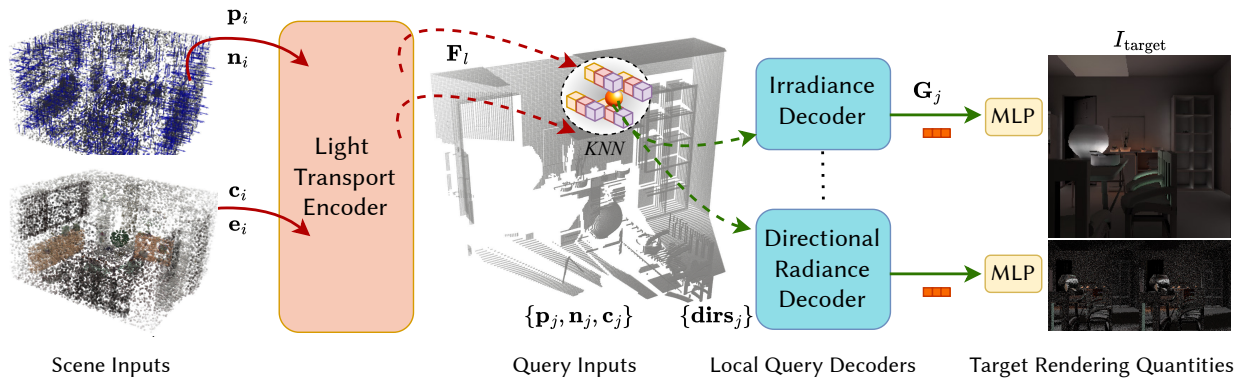


Fig. 3. Our pipeline. Our method first converts the original 3D assets of each scene into a point cloud with associated features (on the left). The scenes are then encoded into a light transport embedding, consisting of latent codes (depicted as small blocks within the dashed circle) anchored at the scene point positions. At render time (center), each query point gathers local latent codes via k-nearest neighbors and feeds them together with the query point attributes into a target-specific query decoder, of which the core is a cross-attention.

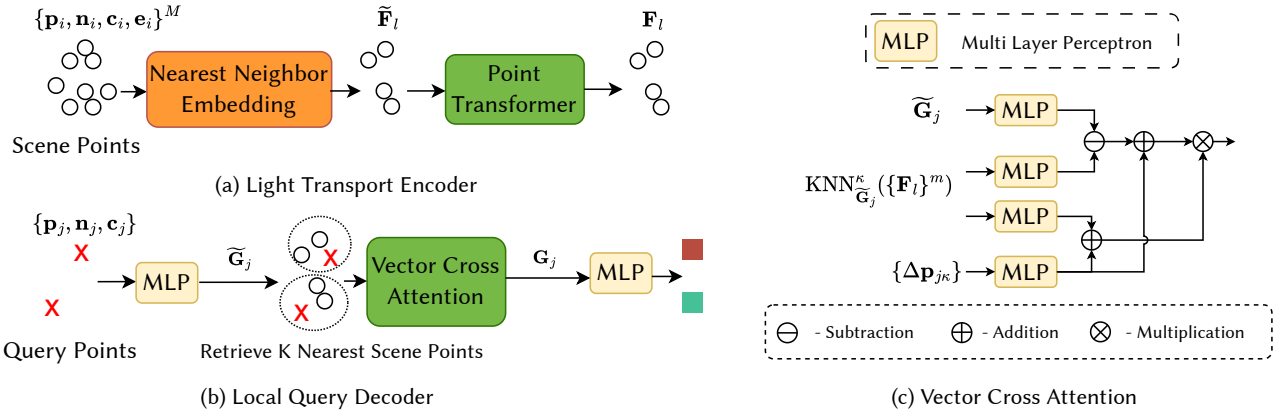


Fig. 4. Network architecture for Light Transport Encoder and Local Query Irradiance Decoder.

Nearest Neighbor Embedding

$$\{\widetilde{\mathbf{X}}_l\}^m = \text{FPS}(\{\mathbf{X}_i\}^M), \ \textit{where}\ \mathbf{X}_i = \mathcal{F}((\mathbf{p}_i, \mathbf{n}_i, \mathbf{c}_i, \mathbf{e}_i)) \quad (5)$$

$$\widehat{\mathbf{X}}_l = \text{concat}\left(\text{KNN}^k_{\widetilde{\mathbf{X}}_l}(\{\mathbf{X}_i\}^M) - \widetilde{\mathbf{X}}_l, \widetilde{\mathbf{X}}_l\right) \quad (6)$$

$$\widetilde{\mathbf{F}}_l = \max_k \mathcal{G}(\widehat{\mathbf{X}}_l) \quad (7)$$

Here, $\mathcal{F}$ is a projection function that converts input points to a latent embedding $\mathbf{X}_i$, and FPS performs farthest point sampling to subsample the original scene points $\{\mathbf{X}_i\}^M$ to m points. Here each application of FPS halves the number of points (i.e., $m = 1/2 * M$), though this reduction factor can be adjusted when needed). After this step, we obtain subsampled point latents $\widetilde{\mathbf{X}}_l$. $\text{KNN}^k_{\widetilde{\mathbf{X}}_l}$ finds k nearest neighbors for each subsampled point $\mathbf{X}_l$ from the original scene, and the difference vectors between the KNN center and its neighbors are computed. These are then concatenated with each center $\mathbf{X}_l$ using concat, followed by another projection function $\mathcal{G}$, and then max-pooled along the k neighbors to create the point embedding $\widetilde{\mathbf{F}}_l$.

As mentioned previously, we model interactions between these scene points using Point Transformer V3 (PTV3, [Wu et al. 2024]) as the backbone, which efficiently encodes any input-point representation:

$$\{\mathbf{F}_l\}^m = \text{PTV3}(\{\widetilde{\mathbf{F}}_l\}^m), \quad (8)$$

where PTV3 refers to the point transformer architecture operating on the derived scene embedding across all points $\{\widetilde{\mathbf{F}}_l\}^m$ to derive the final per-point latent $\{\mathbf{F}_l\}^m$ (dubbed *Light Transport Embedding*). PTV3 introduces point cloud serialization and patch-based attention, replacing costly KNN queries and relative positional encodings. This design enables an efficient expansion of the receptive field up to 1024 points while reducing memory and latency. These properties are particularly suited to our setting, where modeling light transport requires capturing long-range interactions across densely sampled

scenes. The overall operation in this stage is denoted as the Light Transport Encoder in Fig. 4.

4.3 Local Query Decoding

Our light-transport embedding can be repurposed for various downstream rendering tasks simply by swapping in dedicated, task -specific decoders. We show that, after training on a large-scale indoor dataset of diffuse scenes for irradiance prediction (section 6), our model can be quickly fine-tuned—using only limited additional training steps on a smaller dataset with both diffuse and glossy materials–to handle glossy reflections, all while leveraging the pretrained weights (section 7).

Given the Light Transport Embeddings $\{\mathbf{F}_l\}^m$ of a scene, we wish to estimate the global illumination related quantities for any given 3D point in the scene. For view-independent training, we sample the scene geometries; for rendering, we obtain the ray intersection points. These points are defined as *query points* (see section 4.1 and Fig. 4). To do so, we propose a Local Query Decoder, that aggregates information from neighboring scene embeddings to synthesize the final desired output (e.g., irradiance).

Similar to the previous section, we first encode each query point based on its position, normals, and material properties, i.e., $(\mathbf{p}_j, \mathbf{n}_j, \mathbf{c}_j)$ respectively, using an MLP. Using KNN, we obtain the neighboring point embeddings encoded from the scene inputs. For each query point, not all points in its immediate neighborhood contribute equally—thus, performing learned aggregation is essential. We also verify this experimentally in Section 8.

In accordance with the theme of this paper, we leverage a transformer to iteratively aggregate features from neighborhood scene points onto the query point. Rather than applying the standard dot-product attention between the query and scene points (as query and key, values respectively), we use a vector cross-attention operation. Specifically, we replace the dot product attention with subtraction as the relation function. Compared to the dot product, which collapses the feature dimension into a scalar, subtraction attention computes distinct attention scores for each channel of the value matrix, increasing diversity in feature interactions [Fan et al. 2022; Zhao et al. 2021].

Additionally, we augment the attention and value matrices with the relative distances between each query center and its KNN neighbors, denoted by $\Delta\mathbf{p}_{j\kappa}$. Formally, we write:

$$\widetilde{\mathbf{G}}_j = \mathcal{H}(\mathbf{p}_j, \mathbf{n}_j, \mathbf{c}_j), \mathbf{KV} = \mathrm{KNN}^{\kappa}_{\widetilde{\mathbf{G}}_j}(\{\mathbf{F}_l\}^m) \tag{9}$$

$$\mathbf{P}_{j\kappa} = \gamma(\Delta\mathbf{p}_{j\kappa}), \Delta\mathbf{p}_{j\kappa} = \mathbf{p}_j - \mathbf{p}_\kappa \tag{10}$$

$$\mathbf{G}^q_j = \mathrm{W}_q(\widetilde{\mathbf{G}}_j), \mathbf{G}^k_j = \mathrm{W}_k(\mathbf{KV}), \mathbf{G}^v = \mathrm{W}_v(\mathbf{KV}) \tag{11}$$

$$\mathbf{A} = \mathbf{G}^q_j - \mathbf{G}^k_j + \mathbf{P}_{j\kappa} \tag{12}$$

$$\mathbf{G}_j = \mathrm{sum}_\kappa(A(\mathbf{G}^v + \mathbf{P}_{j\kappa})) \tag{13}$$

Here, $\mathcal{H}$ represents a projection function that converts query point input into a feature $\widetilde{\mathbf{G}}_j$, and $\mathrm{KNN}^{\kappa}_{\widetilde{\mathbf{G}}_j}$ retrieves κ nearest neighbors for each query point from the scene latent codes. γ is another projection function that maps the relative distance vector $\Delta\mathbf{p}_{j\kappa}$ to a higher-dimensional representation $\mathbf{P}_{j\kappa}$. W_q, W_k and W_v project the corresponding query, key, and value inputs in the vector cross-attention operation. The final latent embedding for each query point given by $\mathbf{G}_j$ is obtained by aggregating the value features using the computed attention values $\mathbf{A}$ through summing up along κ neighboring points (see fig. 4(b) and (c)).

The final output $\mathbf{I}_{\text{out}}$ is derived as follows:

$$\mathbf{I}_{\text{out}} = \mathcal{W}_{\text{out}}(\{\mathbf{G}^N_j\}), \tag{14}$$

where $\mathcal{W}_{\text{out}}$ represents the final output projection to convert the set of query points $\{\mathbf{G}^N_j\}$ to the required rendering quantities. These projections are implemented as multi-layer perceptrons (MLPs).

5 Implementation

Training dataset. We curated a large-scale indoor global illumination dataset to be publicly released with all code, preprocessing scripts, and trained model weights to support reproducibility and future research (see Figure 5 for an overview). Built upon the procedural generation rules from Raistrick et al. [2024], it comprises 13,900 Blender scenes spanning a wide variety of floorplans and asset collections: furniture, appliances, cookware and dining utensils, decorative objects, and architectural elements. We refined each scene by removing low-quality meshes and duplicate elements (e.g., overlapping exteriors or carpeting) to minimize confusion during training. We also implemented heuristic-driven, randomized placement of rectangular area lights flush with the ceiling. For each room, we detect the ceiling plane and sample positions within its bounding polygon, maintaining clearance from walls and large objects to emulate realistic overhead fixtures. Each scene is exported to PBRT format, with high-quality triangle meshes and baked high-resolution textures for each asset. The resulting files range from 500 MB to 2 GB. For ground-truth irradiance computation, we sample 2 million query points per scene and trace 1,024 stratified, cosine-weighted rays per point with a maximum depth of five. Full dataset generation, including mesh processing, texture baking, and rendering, required approximately 250 CPU/GPU days on our compute cluster. Additionally, we created a smaller dataset for preliminary training of spatial-directional radiance fields, where glossy surfaces are modeled as conductors using the GGX microfacet distribution [Walter et al. 2007].

Training scheme. To avoid dependence on specific camera viewpoints, we train directly on query points uniformly sampled across the entire scene—roughly 2 million points per scene. We train on 90% of the 12k scenes and hold out the rest to test the generalizability. Our model is trained on four A10 GPUs for approximately five days, using learning rate warm-up followed by cosine decay. Each batch contains 8,192 query points, with a per-GPU batch size of 3.

Rendering pipeline integration and timing breakdown. We integrate our PyTorch-based network inference with `Falcor`'s framebuffer [Kallweit et al. 2022] via CUDA interop. For the local query decoder, we extend `cudaKDTree` [Wald 2023] to accelerate k-nearest neighbor search. We report average runtime on our test set: the *Light Transport Encoder* takes 208ms, which is dominated by the transformer and amortized over multiple frames; the *Local Query*

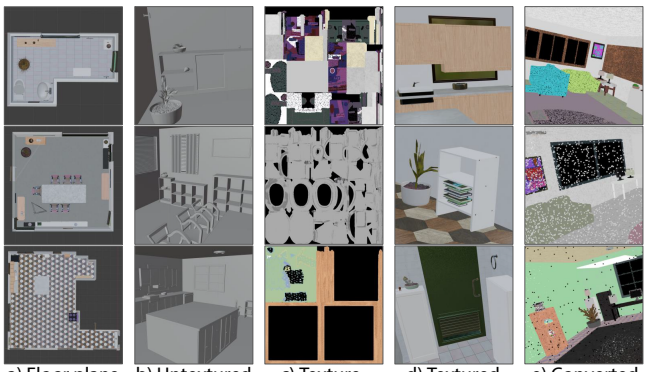


Fig. 5. **Dataset overview.** From left to right, examples of: a) generated floor plans, b) generated untextured meshes for a floor plan, c) generated texture atlases for mesh objects, d) textured mesh objects, e) textured meshes converted to scene points directly fed into our pipeline.

Decoder requires 368ms for generating a 512×512 resolution image on a single NVIDIA RTX 5880 Ada GPU, which includes 25-36ms for GPU KNN search with the rest spent on attention. Since our model is view-independent, the encoding cost can be efficiently amortized across multiple frames. Although we have not achieved real-time framerates yet, fusing attention layers into a custom CUDA kernel and model distillation are promising avenues for future optimization.

Loss function. We optimize the entire network end-to-end to minimize the relative L2 loss in log space, both when estimating irradiance and radiance [Müller et al. 2021]:

$$\mathcal{L} = \frac{1}{|N|} \sum_{\mathbf{q} \in O} \left(\frac{\log(\hat{y}(\mathbf{q}) + 1) - \log(y_{\text{gt}}(\mathbf{q}) + 1)}{\log(\hat{y}(\mathbf{q}) + 1) + \epsilon} \right)^2,$$

where $y_{\text{gt}}(\mathbf{q})$ is the ground truth value for $\mathbf{q}$ and $|N|$ is the number of query points.

6 Application: Diffuse global illumination

Baselines. To our knowledge, there is no established baseline in neural global illumination that targets cross-scene generalization using only 3D scene configurations. To evaluate our method in this setting, we compare against the following representative baselines:

Path tracing serves as a reference for physically accurate rendering. Assuming a typical real-time path tracer runs at 60 FPS, we use 64 samples per pixel for comparison, as our inference time is less than one second. We want to stress that the goal here is not to beat path tracing. While not yet a drop-in replacement for path tracing, our model benefits from ongoing academia and industry optimizations, suggesting integration in the near future may be feasible.

Deep shading [Nalbach et al. 2017], one of the first works to demonstrate learning-based global illumination. It takes as input screen-space G-buffer attributes (such as positions, normals, and material properties) along with direct lighting, and predicts global illumination as a post-process. While it demonstrates that neural networks can approximate certain light transport effects, it operates purely in 2D screen space, making it inherently view-dependent and lacking true 3D spatial understanding. We train their model on our camera-view renderings and additionally provide direct lighting to better match their setup. However, unlike theirs, our pipeline uses raw light source points and does not rely on any rasterized or path-traced illumination cues.

Hermosilla et al. [2019]'s work is the closest to our setup but is limited to single-object scenes with environment lighting and lacks support for spatially varying textures. The original model struggles with the complexity and scale diversity of our dataset, so we replace its radius-based furthest point sampling, used for progressive downsampling, with a target-count approach that reduces the point cloud to fixed sizes across levels. We also expand feature dimensions eightfold to increase model capacity. To reduce GPU memory usage during training, we adopt the memory-efficient summation order proposed in the work by Wu et al. [2019].

We present qualitative results on six diverse floor plans in Figure 6. To better highlight indirect illumination quality, the images show irradiance without texture modulation to avoid visual distraction. Figure 10 on the other hand includes a fully rendered image with texture and direct lighting to illustrate practical applicability. In all other cases, visible colors originate purely from indirect lighting. The HALLWAY and LIVING ROOM setups are especially challenging for a path tracer, as the light source is placed in a distant corner, causing parts of the room to be occluded from the main illuminated area. Our method faithfully captures color bleeding from multi-bounce path tracing and soft shadowing. Minor artifacts remain, such as light leaks at the base of the toilet in BATH ROOM and beneath the bowl in DINING ROOM 1, as well as slight color shifting in LIVING ROOM. Quantitative results on the test set are provided in table 1.

The deep shading method [Nalbach et al. 2017] is confined to screen-space and has limited awareness of the actual 3D scene. As a result, it struggles to capture effects caused by out-of-sight geometries, exhibits view inconsistencies, and generalizes poorly to unseen scenes. To expose its limitations, we include the test scenes in the training set and show the resulting overfitted predictions in Figure 6. Results on unseen test scenes are provided in the supplemental. For instance, accurately capturing the shadow cast by a light source onto the ceiling (indirect light bouncing from the floor) in DINING ROOM 2 is not feasible without access to the 3D knowledge. By using direct lighting as an illumination cue, it was able to predict the area above the light source to be dark, while the region directly beneath it appears brightest (LIVING ROOM). Recent advances in video foundation models may offer improved spatial reasoning by leveraging the 2D+time dimension. However, with the increasing availability of large-scale 3D data, it may be more effective to bypass the indirect route of reconstructing 3D from 2D projections and instead embrace direct 3D representations—enabling full editability and consistent understanding, as demonstrated by our method.

In general, we observe that transformers exhibit strong generalization ability and are surprisingly resistant to overfitting, even on small datasets. In contrast, Hermosilla et al. [2019] is built upon point convolutions, which struggle to learn effectively from our complex dataset. Although we modified their model to better match our scene complexity, it still fails to capture the intricate nature

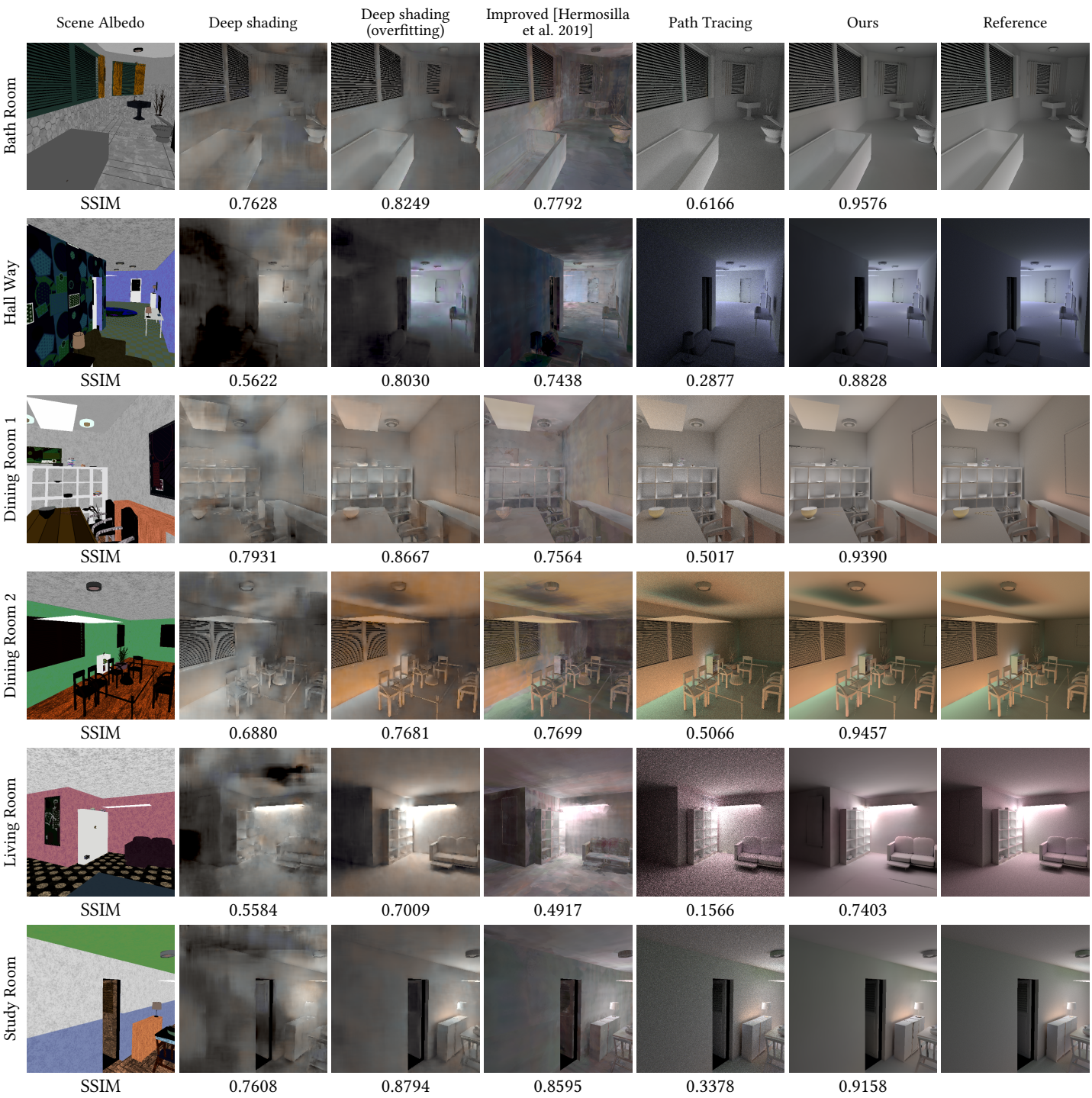


Fig. 6. Irradiance prediction comparison with baseline representatives across six diverse floor-plans under varying illumination conditions. We primarily highlight the substantial qualitative visual improvements while showing SSIM as a quantitative reference. We evaluate our model on a held-out test set to demonstrate its generalization capability. Notably, no per-scene training is performed—unlike prior neural global illumination approaches. For Deep Shading, which fails to generalize to new scenes, we include the test scenes in their training set to show their **overfitting** results here. Additionally, we extend the original implementation by Hermosilla et al. [2019] to handle our more complex scenes, as their original setup is limited to simple object-level inputs and performs poorly on our dataset (see section 6). The first column shows the corresponding albedo to provide readers with context for the color of the indirect lighting, while we avoid showing albedo-modulated images to maintain a clear focus on the quality of the predicted irradiance without distraction.

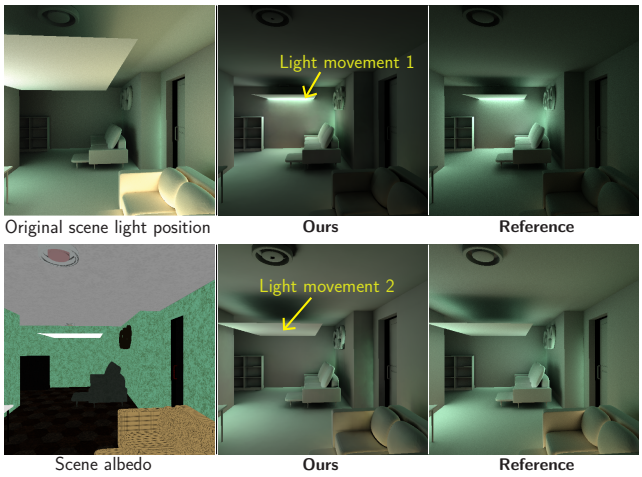


Fig. 7. As the area light moves away or toward the camera, both the shadow and indirect illumination respond accordingly. The model generalizes reasonably well to novel light placements within the same scene. We expect that increasing the diversity of light positions during training would further enhance the quality and robustness of the results. Again, we show the corresponding albedo to provide readers context for the color of the indirect lighting, while we avoid showing albedo-modulated images to maintain a clear focus on the quality of the predicted irradiance without distraction.

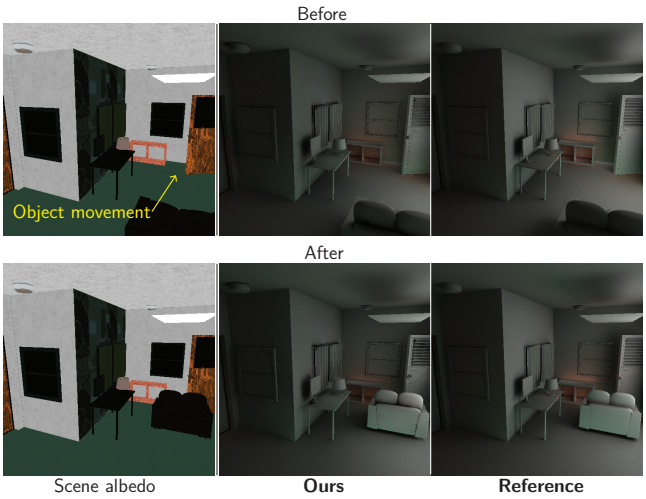


Fig. 8. Irradiance prediction under non-emissive object movement. Unlike previous works, no per-scene training is performed. As the sofa moves toward the light source, it receives stronger indirect illumination, while nearby regions—such as the cabinet near the wall—become darker due to increased occlusion. Again, we show albedo separately to maintain a clear focus on the quality of the predicted irradiance without distraction.

of global illumination and suffers from noticeable artifacts. We believe transformers offer an advantage due to their ability to model long-range dependencies and dynamically aggregate features across spatially distant regions, which is critical for accurately representing light transport.

While path tracing often produces noisy outputs, these are typically refined using post-processing denoisers in modern pipelines. We do not claim our method replaces the full path tracing plus

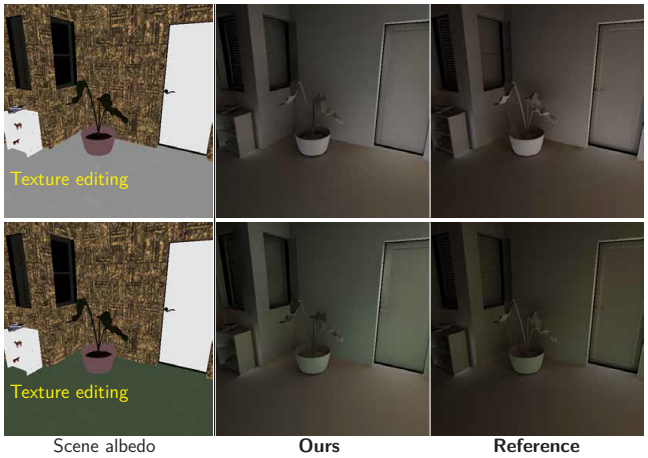


Fig. 9. Irradiance prediction under material editing. When we change the floor's albedo from a pale (whitish) tone to a greenish hue, the indirect lighting cast onto the walls adjusts correctly. No per-scene retraining is needed.

denoising setup, but instead emphasize the potential of predicting global illumination directly from 3D scene configurations. Unlike traditional path tracing, where shader execution diverges across pixels due to bounce paths, occlusion, and material interactions, our method performs a uniform forward pass per pixel. This regularity enables efficient parallel execution on modern GPUs and offers predictable performance across varying scene complexity. It also replaces the irregular, memory-bound nature of path tracing with more streamlined and predictable memory access. As power efficiency becomes increasingly critical in modern architectures, transformer-based inference presents a promising direction for scalable light transport.

View consistency. Our model is inherently view-independent (section 5), as it is trained on query points sampled directly from the 3D scene rather than camera-based observations. Please refer to our supplemental video.

Light source movement. We assess the model's robustness to changing light positions, as illustrated in fig. 7. In this living room example, moving the area light toward the camera leads to corresponding changes in shadows and indirect illumination. The shadow shifts relative to the light source due to occlusion of indirect light reflected from the floor, and the two sofa areas become brighter or darker as a result. Although each scene was trained with a fixed light configuration, the model not only generalizes to new floor plans and geometries, but also responds robustly to novel light placements. This behavior reflects the strength of the learned 3D light transport embedding in capturing global illumination dynamics. We expect that training on a wider range of lighting conditions would further enhance both accuracy and robustness.

Object movement. We evaluate the model's robustness to non-emissive object movement, as shown in fig. 8. In this example, as the sofa moves closer to the light source, it becomes more brightly lit, while the small cabinet near the wall appears darker due to

Metric	Deep Shading (overfitting)	Improved [Hermosilla et al.]	Path Tracing	Ours
MSE ↓	0.071	0.171	0.051	0.048
SSIM ↑	0.882	0.775	0.425	0.912

Table 1. Quantitative comparison across methods on 105 random scenes from our test set. Note that the Deep shading result is **overfitting** to evaluated camera views since it struggles to generalize to novel scenes within the diverse dataset. Our visual quality is overall significantly better than the baselines. We show MSE and SSIM for comparison.

increased occlusion. As expected, the shadow consistently follows the moving object.

Material editing. We evaluate the model's response to material editing. In fig. 9, changing the floor's albedo from pale white to deep green causes the predicted irradiance field to update accordingly, modifying the indirect lighting on nearby walls. This result is achieved without any per-scene retraining. Expanding the dataset with a wider range of textures would likely improve robustness in more diverse editing scenarios.

7 Application: Spatial-directional incoming radiance field via neural basis

The irradiance prediction task highlights our model's ability to generalize across diverse scenes by capturing global illumination effects such as color bleeding, soft shadows, and multi-bounce indirect lighting. As a view-independent quantity, irradiance evaluates the model's core capability to encode global interactions, laying the groundwork for extending to more complex, view-dependent glossy materials. Hence, we further generalize our approach by re-purposing our pre-trained light transport encoder and directly predicting a spatial-directional radiance field, which can later be combined with any BRDF model.

We construct a smaller dataset for preliminary experiments by randomly moving objects within the rectangular living room floorplan and supporting glossy materials using the GGX microfacet distribution (see Figure 11). We add one additional material attribute, roughness and we augment the irradiance input with 1024 hemispherical directions to predict directional radiance. Ground truth is path-traced using cosine-weighted sampling, discretized into 1024 (32×32) directional bins with 64 samples per bin.

We fine-tune from our pre-trained encoder from the irradiance prediction task to quickly converge on this new task. This demonstrates how our shared encoder can be reused across tasks by attaching multi-head, task-specific decoders to predict various downstream rendering quantities. Radiance field fitting results are shown in Figure 11b.

In parallel, we evaluate our neural directional basis against spherical harmonics (SH). While SH provides a compact encoding, its global polynomials under-represent high-frequency glossy structure: low degrees oversmooth highlights, and higher degrees $(l_{\max}+1)^2$ tend to introduce ringing artifacts. By conditioning on scene geometry and material context, our learned basis achieves significantly higher fidelity for complex incident-radiance distributions. Qualitative comparisons and error vs. degree $l_{\max}$ are reported in fig. 12, where our method preserves sharp detail without ringing and attains lower reconstruction error.

While the two previous applications demonstrate our model's ability to generalize across scenes by directly predicting global illumination, we further demonstrate that our approach can also support unbiased rendering by aiding importance sampling in traditional path tracing pipelines.

Jump-starting Path Guiding. Building on the predicted directional radiance field from Section 7, we explore its use in path guiding initialization. As shown in Figure 13, we compare standard BRDF sampling with importance sampling derived from product of our model's predicted radiance field and the BRDF term via CDF inversion. At low sample counts (e.g., 2 samples per pixel), modern path guiding methods ([Vorba et al. 2019], [Herholz et al. 2025])—which rely on per-scene optimization — struggle to accumulate enough samples to construct a usable distribution. In contrast, our learned radiance field can serve as a high-quality initializer, effectively "jump-starting" the guiding process.

Our model is not a standalone path guiding method, but rather serves as an initialization for established methods that rely on defensive BRDF sampling. It predicts a radiance distribution before path guiding methods have gathered enough samples (typically tens to hundreds). This approach draws parallels with meta-learning, where knowledge accumulated across scenes enables fast adaptation to new instances—in our case, facilitating efficient light transport simulation without per-scene retraining. Note that existing methods aren't directly comparable here, as they do not operate well in such low-sample cases.

8 Ablation studies and Analysis

We have made several specific design choices in terms of training data generation and network architectures. In what follows, we present ablation studies on those design choices and analysis of those choices including some important hyperparameter selection.

Network design ablation. We verify our design choices and detail our findings in Figure 14, which also serves as a conceptual roadmap of our modeling and experimental process. The baseline Vanilla model uses a basic global transformer as an encoder. For each query point, it retrieves neighboring scene points via k-nearest neighbors (KNN), followed by simple pooling and multi-layer perceptron as the decoder to obtain the rendered output. Although this ablation shows promising results, it is extremely challenging to train the model which suffers from severe scalability issues due to the quadratic complexity of global attention, making it inefficient for high-resolution point clouds. To address this, our next baseline replaces the Transformer with the proposed Light Transport Encoder, which is significantly more scalable and better suited for complex 3D scenes. It still performs a naïve KNN aggregation at the decoding stage. Surprisingly, this configuration results in slightly worse performance compared to the Vanilla baseline. Despite this drop, we prioritize a faster and more scalable architecture type. Our full model completes the pipeline by incorporating the Local Query Decoder, which introduces localized, learnable aggregation at the query points. This component effectively recovers the quality lost in

Fig. 10. Full renderings with direct illumination and our predicted global illumination for diverse floorplans, lighting conditions and camera views.

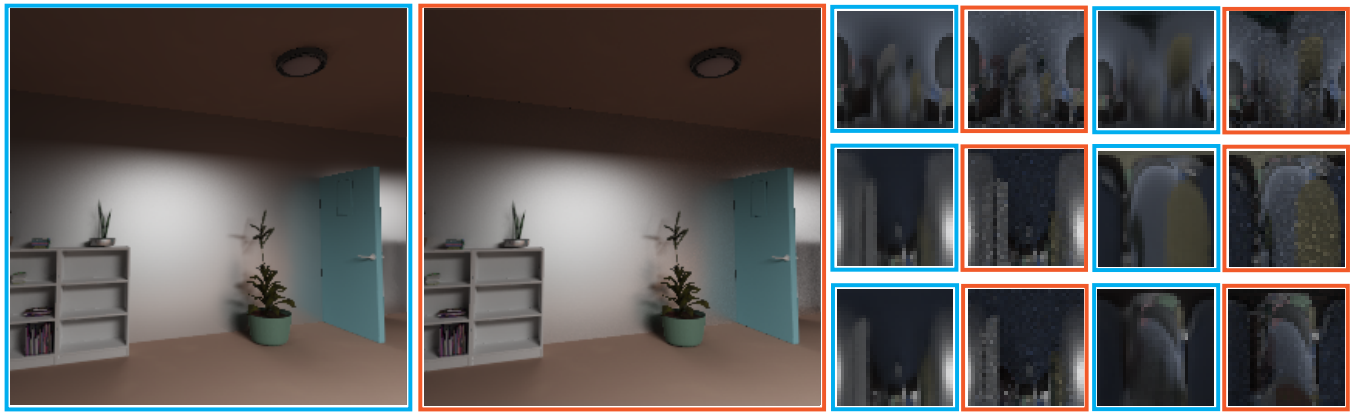
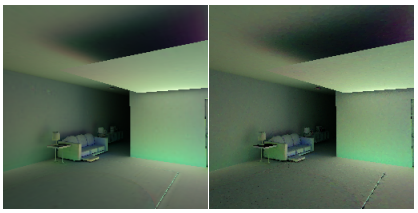


Fig. 11. Rendering comparison for a mixed scene with diffuse and glossy materials. The two boxes use a conductor BRDF modeled with a GGX microfacet distribution and a roughness of 0.1. For glossy surfaces, our Directional Radiance Decoder directly predicts incident radiance given a shading point and any incoming direction. Global illumination in the left uses BRDF integrated with our predicted directional radiance field, and the middle one shows a path traced reference (1024 directional bins per point and 64 samples per bin). The slices on the right visualize the radiance field at 10 randomly selected shading points, comparing our predicted values to path-traced reference over 1024 hemispherical directions.

the previous stage and achieves the best overall performance across metrics. Together, these experiments highlight the importance of both encoding and decoding design choices, and the ablation study reinforces that each component contributes meaningfully to the final model's effectiveness and efficiency.

Spatial neural primitives. We chose to use point-based neural primitives as the intermediate representation for 3D embedding, where the latent codes are anchored at the scene points. This point-anchor format is naturally suited for the multi-scene encoding via transformers and cross-scene generalization. Several other neural graphics primitives have been proposed for compact scene representation and we acknowledge that there can be potentially better choices. We conducted a preliminary experiment to show that in the single-scene optimization settings, the point-anchored latent code displacements may not match the efficiency of multi-resolution hash encoding [Müller et al. 2022]. As shown in the figure on the right, we make our per-point latent codes trainable and use a MLP decoder to regress a single scene.



Extending our encoder to integrate with more expressive neural primitive representations remains an interesting direction for future work. For instance, extending our encoder to produce latent features that index into a multi-resolution hash grid [Müller et al. 2022] could potentially combine the benefits of scene-level generalization with high-resolution spatial encoding and faster inference.

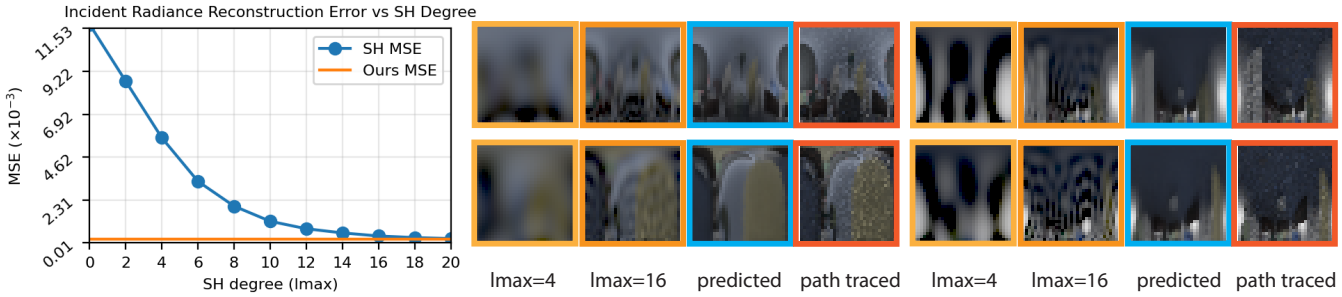


Fig. 12. To assess the effectiveness of our scene encoder and neural basis, we compare our incident-radiance predictions against spherical-harmonics (SH) reconstructions at four randomly sampled surface points. Our method preserves high-frequency glossy reflections. In contrast, low-degree SH yields overly blurred results (first column), while high-degree SH (second column) introduces ringing (Gibbs-like) artifacts. The left plot reports the reconstruction error (MSE) as the SH degree l_{max} increases.

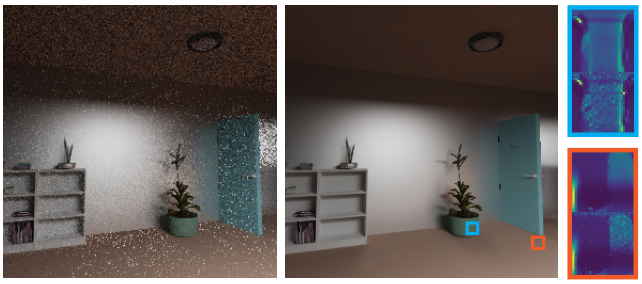
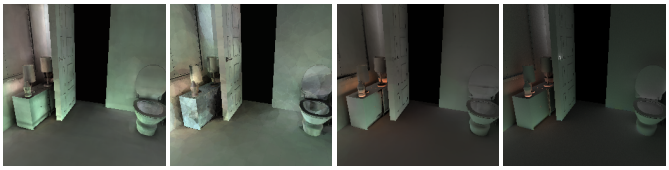


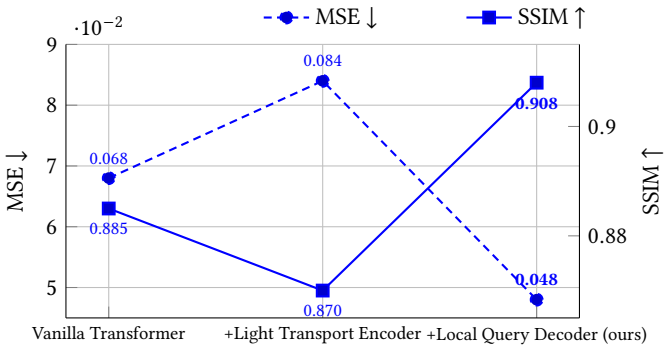
Fig. 13. Path tracing results using different sampling strategies. Left: BRDF sampling with 2 samples per pixel (spp); middle: importance sampling via CDF inversion based on the normalized histogram of the product of our predicted incoming radiance field and brdf values; right: the predicted (top) and path traced (bottom) sampling distributions for the incident radiance integrand at two random points. The reference can be seen in fig. 11. Our model-driven sampling significantly reduces noise in low-sample regimes. As more samples are accumulated, our predicted PDF can serve as an effective initializer for modern path guiding methods, mitigating the cold-start problem inherent in per-scene optimization approaches. Note that the cosine term is included in both of the importance sampling strategies.

Scene point sampling density and sampling strategy. We follow the work by Hermosilla et al. [2018]; Li et al. [2019] to account for the importance sampling densities of points but find their impact on performance negligible. We observe that uniform sampling of 20k points per scene performs comparably to simple heuristics such as biasing toward smaller objects. In general, we find that the effect of sampling strategies diminishes once the scene is sufficiently covered. While performance is not highly sensitive to sampling strategy, the sampling density should still reflect geometric and texture frequencies to capture the signal accurately.

Number of query points for view-independent training. To inform our design choices, we perform a quick verification experiment illustrated in Figure 15. In this single-scene irradiance overfitting setup, we omit the encoder, make the point-based latent codes trainable, and use a simple MLP as the decoder. The experiment evaluates the effect of varying the number of query points. We find that using 2 million query points provides sufficient coverage for complex scenes, offering a good trade-off between performance and memory usage. For simpler scenes, this number can be significantly reduced; for instance, 200k points are sufficient for the Cornell Box.



(a) Qualitative results for ablating different components of the model. This ablation also serves as a **roadmap** of our design and experimental journey. **Left:** Global vanilla Transformer encoder—an intuitive starting point, but quadratic attention makes it non-scalable with point count and hard to train. **Middle:** Our Light Transport Encoder (section 4.2) with naïve KNN aggregation: highly scalable and fast, but with a noticeable quality drop. **Right:** Adding the Local Query Decoder (section 4.3) completes the design and recovers quality, achieving the best overall performance.



(b) Quantitative complement (100 random test scenes) aligned to the same roadmap. The x-axis follows the design stages used in the qualitative figure: *Vanilla* → *Light Transport Encoder* → *Ours*. MSE (left; lower is better) and SSIM (right; higher is better) are plotted simultaneously. Our full model achieves the lowest MSE and highest SSIM.

Fig. 14. Ablation roadmap on the components of our model with separate qualitative (top) and quantitative (bottom) views using the same stage ordering.

262k query points 2M query points Reference

Fig. 15. We analyze performance under varying numbers of sampled query points on the scene geometry in single-scene overfitting setting, where the model predicts irradiance at primary ray hits.

K	PSNR↑	**# scene points**	PSNR↑
8	26.96	5k	28.51
32	34.57	10k	30.94
48	34.71	20k	34.57

Table 2. Ablation results for the number of neighbors (k) during decoding, and number of input scene points.

More network hyper-parameters. We further ablate on the number of input scene points and the value of k for the KNN search in section 4.3. We conduct these experiments on a small subset of the data, training on 90 scenes and testing on 10 scenes for easy analysis. From Table. 2, we see that a smaller value for k (=8) yields significantly lower performance and eventually saturates at higher values of k (> 32). Larger k values also incur much higher computational costs, so we set k=32 as a balanced choice for our best model. As expected, increasing the number of input scene points directly improves the performance, and our design choices are made with this in consideration (e.g. efficient light transport embedding section 4, local decoding section 4.3). For the sake of this paper, we choose to train on 20k input scene points, and further improvements can include scaling this up further with more scenes and more dense input points.

9 Limitation

Failure cases and artifacts. We observe color shifting in some scenes, likely because the color tones of some textures are less seen in the dataset. In general, we found that when the original color tone is intense, our predicted color is slightly lighter (see Hallway, Livingroom, and Study room in Figure 6 and Figure 9). There are also some light leaks around smaller objects and failures to capture some intricate indirect highlights. For example, the artifact on the floor in the 5th scene of Fig. 6 is caused by the inclusion of scene points sampled beneath the carpet, which are completely black due to occlusion. As a result, the edge region of the carpet receives latent codes from both black and valid regions, causing inconsistency. Increasing the number of neighbors k helps resolve this issue — we found that using k=64 removes the artifact and is more robust to the potential issues of the rendered dataset. Alternatively, this issue can be avoided by excluding fully occluded regions when sampling scene points.

Fig. 16. **Two types of failure cases.** We observe occasional color shifting in a small number of scenes (one example on the left), likely due to under-represented texture color tones in the training set. Increasing the number of training scenes with more diverse textures could help mitigate this issue. Another failure case involves particularly challenging lighting conditions, such as the example on the right, where the light source is located in a separate room, creating an "ajar" lighting configuration. Such cases are rare in the current dataset. We conjecture that with more examples of ajar scenes included during training, the model would better handle these scenarios.

As with most deep learning approaches, our model is designed to generalize within the domain it was trained on. Completely out-of-distribution scenes might introduce a significant domain shift. To address this, two natural extensions would be: (1) expanding the training dataset to include more diverse scene types, e.g. outdoor scenes, and (2) applying limited fine-tuning to adapt the model to new domains.

10 Conclusion and future work

We introduced a transformer-based framework for learning a generalizable 3D light transport embedding that predicts global illumination directly from scene geometry, material properties, and lighting configurations. By capturing long-range spatial interactions through attention, our method produces view- and resolution-independent results across a wide variety of scenes without requiring per-scene retraining or any rasterized or ray-traced illumination cues.

We showed that our light transport embedding can be repurposed with limited fine-tuning for a range of downstream applications, including directional radiance field prediction for glossy materials and guided importance sampling for unbiased rendering. This flexibility points to a broader vision of a unified, neural representation of light transport that can adapt to many rendering tasks.

Future work includes extending the method to support larger-scale environments, such as multi-room interiors or outdoor scenes, and incorporating participating media to handle effects like fog, smoke, and translucency. Finally, we note that, unlike a traditional path tracing pipeline, our model is differentiable by construction, making it possibly well-suited as a building block in inverse rendering, scene reconstruction, and differentiable rendering systems. While far from being an off-the-shelf replacement for classical rendering pipeline, this approach highlights the potential for promising use of tensor cores in place of RT cores.

Acknowledgments

This work was support in part by NSF grants 2105806, 2212085 and the Ronald L. Graham Chair. We also acknowledge gifts from Adobe, Google, Qualcomm and Rembrand, and the UC San Diego Center for Visual Computing. The authors would like to thank Aaron Lefohn and Chris Wyman for their support; Matt Pharr for valuable input and discussions along the way; Thomas Akenine-Möller, Jacob Munkberg, Jon Hasselgren and Zian Wang for their suggestions

regarding the evaluation, dataset and training clusters. Bing owes particular thanks to Alex Trevithick for pointers to procedural scene generation and many helpful discussions; Zimo Wang and Nithin Raghavan for generously sharing cluster training quotas; Yang Zhou for proofreading and comments; and her fellow interns for helpful discussions in the early stage.

References

James Arvo, Kenneth Torrance, and Brian Smits. 1994. A framework for the analysis of error in global illumination algorithms. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. 75–84.

Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony DeRose, and Fabrice Rousselle. 2017. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Transactions on Graphics* 36, 4 (2017), 97:1–97:14. https://doi.org/10.1145/3072959.3073708

Chakravarty R Chaitanya, Anton S Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, Timo Aila, and Nima Khademi Kalantari. 2017. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 98.

Per H Christensen, Wojciech Jarosz, et al. 2016. The path to path-traced movies. *Foundations and Trends® in Computer Graphics and Vision* 10, 2 (2016), 103–175.

Stavros Diolatzis, Julien Philip, and George Drettakis. 2022. Active Exploration for Neural Global Illumination of Variable Scenes. *ACM Transactions on Graphics* 41, 5 (2022), 171:1–171:18. https://doi.org/10.1145/3522735

SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. 2018. Neural scene representation and rendering. *Science* 360, 6394 (2018), 1204–1210.

Zhiwen Fan, Tianlong Chen, Peihao Wang, and Zhangyang Wang. 2022. CADTransformer: Panoptic Symbol Spotting Transformer for CAD Drawings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10986–10996.

Duan Gao, Haoyuan Mu, and Kun Xu. 2023. Neural Global Illumination: Interactive Indirect Illumination Prediction Under Dynamic Area Lights. *IEEE Transactions on Visualization and Computer Graphics* 29, 12 (2023), 5325–5341. https://doi.org/10.1109/TVCG.2022.3209963

Michaël Gharbi, Tzu-Mao Li, Miika Aittala, Jaakko Lehtinen, and Frédo Durand. 2019. Sample-based Monte Carlo denoising using a kernel-splatting network. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.

Jonathan Granskog, Fabrice Rousselle, Marios Papas, and Jan Novák. 2020. Compositional neural scene representations for shading inference. *ACM Transactions on Graphics* 39, 4 (2020), 135:1–135:13. https://doi.org/10.1145/3386569.3392475

Miloš Hašan, Fabio Pellacini, and Kavita Bala. 2006. Direct-to-indirect transfer for cinematic relighting. *ACM transactions on graphics (TOG)* 25, 3 (2006), 1089–1097.

Miloš Hašan, Fabio Pellacini, and Kavita Bala. 2006. Direct-to-indirect transfer for cinematic relighting. *ACM Trans. Graph.* 25, 3 (July 2006), 1089–1097. https://doi.org/10.1145/1141911.1141998

Sebastian Herholz, Martin Sik, Lea Reichardt, and Marco Manzi. 2025. Path Guiding in Production and Recent Advancements. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Courses (SIGGRAPH Courses '25)*. Association for Computing Machinery, New York, NY, USA, Article 14, 5 pages. https://doi.org/10.1145/3721241.3733994

Pedro Hermosilla, Sebastian Maisch, Tobias Ritschel, and Timo Ropinski. 2019. Deep-learning the Latent Space of Light Transport. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 207–217.

Pedro Hermosilla, Tobias Ritschel, Pere-Pau Vázquez, Àlvar Vinacua, and Timo Ropinski. 2018. Monte carlo convolution for learning on non-uniformly sampled point clouds. *ACM Transactions on Graphics (tog)* 37, 6 (2018), 1–12.

Wojciech Jarosz, Craig Donner, Matthias Zwicker, and Henrik Wann Jensen. 2008. Radiance Caching for Participating Media. *ACM Transactions on Graphics* 27, 1 (March 2008), 7:1–7:11. https://doi.org/10.1145/1330511.1330518

James T Kajiya. 1986. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 143–150.

Simon Kallweit, Petrik Clarberg, Craig Kolb, Tom'aš Davidovič, Kai-Hwa Yao, Theresa Foley, Yong He, Lifan Wu, Lucy Chen, Tomas Akenine-Möller, Chris Wyman, Cyril Crassin, and Nir Benty. 2022. The Falcor Rendering Framework. https://github.com/NVIDIAGameWorks/Falcor https://github.com/NVIDIAGameWorks/Falcor.

Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. 2019. PointConv: Deep Convolutional Networks on 3D Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9621–9630.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics* 41, 4 (2022), 102:1–102:15. https://doi.org/10.1145/3528223.3530127

Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. 2021. Real-time neural radiance caching for path tracing. *ACM Transactions on Graphics* 40, 4 (2021), 36:1–36:16. https://doi.org/10.1145/3450626.3459812

Oliver Nalbach, Elena Arabadzhiyska, Dushyant Mehta, H-P Seidel, and Tobias Ritschel. 2017. Deep shading: convolutional neural networks for screen space shading. In *Computer graphics forum*, Vol. 36. Wiley Online Library, 65–78.

Shree K Nayar, Gurunandan Krishnan, Michael D Grossberg, and Ramesh Raskar. 2006. Fast separation of direct and global components of a scene using high frequency illumination. In *ACM SIGGRAPH 2006 Papers*. 935–944.

OpenAI. 2024. Video generation models as world simulators. https://openai.com/index/video-generation-models-as-world-simulators. Accessed: 2025-05-10.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017a. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 652–660.

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 30. 5099–5108.

Nithin Raghavan, Yan Xiao, Kai-En Lin, Tiancheng Sun, Sai Bi, Zexiang Xu, Tzu-Mao Li, and Ravi Ramamoorthi. 2023. Neural Free-Viewpoint Relighting for Glossy Indirect Illumination. In *Computer Graphics Forum*, Vol. 42. Wiley Online Library, e14885.

Gilles Rainer, Adrien Bousseau, Tobias Ritschel, and George Drettakis. 2022. Neural Precomputed Radiance Transfer. *Computer Graphics Forum* 41, 2 (2022), 365–378. https://doi.org/10.1111/cgf.14480

Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, et al. 2024. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21783–21794.

Ravi Ramamoorthi et al. 2009. Precomputation-based rendering. *Foundations and Trends® in Computer Graphics and Vision* 3, 4 (2009), 281–369.

Peiran Ren, Jiaping Wang, Minmin Gong, Stephen Lin, Xin Tong, and Baining Guo. 2013. Global illumination with radiance regression functions. *ACM Transactions on Graphics* 32, 4 (2013), 130:1–130:12. https://doi.org/10.1145/2461912.2462009

Yawar Siddiqui, Tom Monnier, Filippos Kokkinos, Mahendra Kariya, Yanir Kleiman, Emilien Garreau, Oran Gafni, Natalia Neverova, Andrea Vedaldi, Roman Shapovalov, and David Novotny. 2024. Meta 3D AssetGen: Text-to-Mesh Generation with High-Quality Geometry, Texture, and PBR Materials. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 9532–9564. https://proceedings.neurips.cc/paper_files/paper/2024/file/123cfe7d8b7702ac97aaf4468fc05fa5-Paper-Conference.pdf

Peter-Pike Sloan, Jan Kautz, and John Snyder. 2002a. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph.* 21, 3 (July 2002), 527–536. https://doi.org/10.1145/566654.566612

Peter-Pike Sloan, Jan Kautz, and John Snyder. 2002b. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics* 21, 3 (2002), 527–536. https://doi.org/10.1145/566654.566612

Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beniot Marcotegui, Francois Goulette, and Leonidas Guibas. 2019. KPConv: Flexible and Deformable Convolution for Point Clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 6411–6420.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

Eric Veach. 1998. *Robust Monte Carlo methods for light transport simulation*. Stanford University.

Thomas Vogels, Fabrice Rousselle, Brian McWilliams, Mark Meyer, Steve Bako, Jan Nova´k, Alex Harvill, and Marc Droske. 2018. Denoising with kernel prediction and asymmetric loss functions. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 124.

Jiří Vorba, Johannes Hanika, Sebastian Herholz, Thomas Müller, Jaroslav Křivánek, and Alexander Keller. 2019. Path Guiding in Production. In *ACM SIGGRAPH 2019 Courses* (Los Angeles, California) *(SIGGRAPH '19)*. ACM, New York, NY, USA, Article 18, 77 pages. https://doi.org/10.1145/3305366.3328091

Ingo Wald. 2023. GPU-friendly, Parallel, and (Almost-)In-Place Construction of Left-Balanced k-d Trees. https://arxiv.org/abs/2211.00120

Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. 2007. Microfacet Models for Refraction through Rough Surfaces. *Rendering techniques* 2007 (2007), 18th.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics* 38, 5 (2019), 146:1–146:12.

Gregory J. Ward and Paul S. Heckbert. 1992. Irradiance Gradients. In *Proceedings of the Eurographics Workshop on Rendering Techniques*. 85–98.

Wenxuan Wu, Zhongang Qi, and Li Fuxin. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9621–9630.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. 2024. Point Transformer V3: Simpler, Faster, Stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. https://doi.org/10.1109/CVPR52733.2024.00463

Hanggao Xin, Shaokun Zheng, Kun Xu, and Ling-Qi Yan. 2022. Lightweight Bilateral Convolutional Neural Networks for Interactive Single-Bounce Diffuse Indirect Illumination. *IEEE Transactions on Visualization and Computer Graphics* 28, 4 (2022).

Bing Xu, Junfei Zhang, Rui Wang, Kun Xu, Yong-Liang Yang, Chuan Li, and Rui Tang. 2019. Adversarial Monte Carlo denoising with conditioned auxiliary feature modulation. *ACM Transactions on Graphics* 38, 6 (2019), 224:1–224:12. https://doi.org/10.1145/3355089.3356547

Chong Zeng, Yue Dong, Pieter Peers, Hongzhi Wu, and Xin Tong. 2025. RenderFormer: Transformer-based Neural Rendering of Triangle Meshes with Global Illumination. In *ACM SIGGRAPH 2025 Conference Papers*.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H.S. Torr, and Vladlen Koltun. 2021. Point Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 16259–16268.

Chuankun Zheng, Yuchi Huo, Hongxiang Huang, Hongtao Sheng, Junrong Huang, Rui Tang, Hao Zhu, Rui Wang, and Hujun Bao. 2024. Neural Global Illumination via Superposed Deformable Feature Fields. In *SIGGRAPH Asia 2024 Conference Papers*. 1–11.

Chuankun Zheng, Yuchi Huo, Shaohua Mo, Zhihua Zhong, Zhizhen Wu, Wei Hua, Rui Wang, and Hujun Bao. 2023. NeLT: Object-Oriented Neural Light Transfer. *ACM Trans. Graph.* 42, 5 (2023).

Attila T. Áfra. 2024. Intel® Open Image Denoise. https://www.openimagedenoise.org. Accessed: 2025-05-10.