

RADIANT: OBJECT-CENTRIC FEED-FORWARD GLOBAL ILLUMINATION

Sarah Zhang*, Zhiyang Dou*, Minghao Guo, Benjamin Tod Jones & Wojciech Matusik

MIT CSAIL

ABSTRACT

Feed-forward neural renderers synthesize global illumination across novel 3D scenes without per-scene optimization or runtime ray tracing. Existing transformer-based renderers tokenize individual triangles, so self-attention scales quadratically with mesh resolution. Yet macroscopic radiative transport is governed by the underlying surface geometry and object arrangements rather than a particular mesh discretization; finer tessellation can thus inflate token counts without changing the underlying transport physics, motivating a discretization-agnostic and object-centric representation. We introduce RADIANT (**RA**diance-based **Inter-Actor Neural Transformer**), an object-centric architecture for feed-forward global illumination from point inputs. By representing each object with a compact budget of 32 surface-patch tokens, RADIANT decouples attention sequence length from polygon counts. In a single network, an inter-object scene transformer models 3D radiative exchange and a view decoder renders images at target camera poses. To capture light transport without materializing $O(N^2)$ bias matrices, invariant form-factor attention embeds geometric distances and surface orientations directly into query-key dot products as $SE(3)$ -invariant inner products weighted by learned area and material gains. For primary visibility without mesh connectivity, camera rays cross-attend scene tokens guided by depth-tested point splats in a shared spherical frame. Across a 1.03M-scene benchmark under matched model capacity, RADIANT reaches 25 dB tone-mapped PSNR in $5\times$ fewer training steps than RenderFormer, and achieves 32.2 dB at 600k steps (+3.3 dB over RenderFormer and +7.6 dB over RenderFormer-V2). We further demonstrate the scalability of RADIANT, which completes a 512^2 forward pass of a 500-object scene in 0.83 s, with computational cost scaling with the number of objects rather than polygons. We will release our code and dataset upon publication.

1 INTRODUCTION

Global illumination effects (such as soft contact shadows, diffuse interreflections, and color bleeding) provide visual cues for spatial depth and surface geometry, governed by the rendering equation (Kajiya, 1986). Classical Monte Carlo path tracing (Kajiya, 1986) and radiosity (Goral et al., 1984) compute physically accurate light transport from surface meshes and material definitions, but tracing rays at runtime remains computationally expensive. Per-scene methods such as neural radiance fields and 3D Gaussian splatting (Mildenhall et al., 2020; Kerbl et al., 2023; Hadadan et al., 2021) avoid runtime path tracing by fitting scene-specific representations, but cannot generalize across new environments. Feed-forward neural renderers (Zeng et al., 2025; 2026) instead predict radiance across unseen scenes in a single forward pass without test-time optimization.

Existing feed-forward renderers such as RenderFormer (Zeng et al., 2025; 2026) tokenize explicit triangle meshes, running self-attention over individual primitives (T triangles). Because full attention scales as $O(T^2)$, latency and GPU memory grow rapidly with mesh resolution. Subsequent variants introduce windowed attention (Zeng et al., 2026) or object-level pooling (Du et al., 2026), but still require explicit triangle-based geometry and compute primitive-level attention during encoding. Point clouds, by contrast, are topology-free and interface directly with depth sensors and

*Equal contribution.

Table 1: **Architectural and computational comparison.** *Input*: input geometric modality. *Triangle attention*: extent of self-attention over individual triangles. *Transport tokens* and *Transport attention*: sequence length and computational complexity of global light exchange (T triangles, N scene entities including walls and lights). *Primary visibility*: mechanism resolving camera-ray intersections. *Steps to 25 dB*: optimizer steps required to attain 25 dB tone-mapped PSNR on the 1.03M-scene benchmark under our matched protocol.

Method	Input	Triangle attention	Transport		Primary visibility	Per-scene optimization	Steps to 25 dB
			tokens	attention			
Neural Radiosity	mesh [†]	—	—	—	ray tracing	required	—
RenderFormer	triangles	all pairs	T	$O(T^2)$	learned	none	80k
RenderFormer-V2	triangles	windowed	T	$O(T^2)$	learned	none	n/r
RenderFormer++	triangles	per object	$8N$	$O(N^2)$	G-buffer	none	—
RADIANT (ours)	point inputs	none	$32N$	$O(N^2)$	point splat	none	16k (5×)

[†]Its neural radiance field needs no meshing, but training and rendering trace rays against the scene geometry. n/r: did not reach 25 dB within 600k steps (Tab. 2).

3D generative models. Prior work has learned light transport on point-based scene representations for static and dynamic indoor scenes (Sanzenbacher et al., 2020), as a direct-to-indirect illumination operator on 3D point clouds (Hermosilla et al., 2019), and as generalizable 3D embeddings for diffuse global illumination (Xu et al., 2026). Building on these point-based light-transport methods, we investigate an object-centric formulation that combines compact surface-patch tokens, scene-level transport, and camera-ray HDR decoding guided by mesh-free primary visibility.

We introduce RADIANT (**RA**diance-based **IN**ter-Actor **NE**ural **T**ransformer), an object-centric architecture that learns feed-forward global illumination directly from point inputs. In radiosity and radiative heat transfer (Goral et al., 1984), energy exchange occurs between finite surface regions and objects regardless of surface tessellation. For fixed surface geometry and materials, macroscopic light transport is the same whether a surface is represented by 50 or 50,000 triangles. By defining attention tokens over a compact set of surface patches per object rather than individual polygons, RADIANT decouples transport sequence length from geometric discretization, providing a unified pipeline for meshes and point clouds.

RADIANT divides computation into 3D scene transport and 2D view decoding. A point encoder first extracts 32 surface-patch tokens per object via point set abstraction, bounding global transport to $O(N^2)$ attention over $32N$ tokens for N scene entities without requiring manifold meshes. An inter-object scene transformer then models radiative transfer among these patch tokens. To incorporate geometric distance and surface orientation without the quadratic memory overhead of explicit pairwise bias matrices, we introduce *invariant form-factor attention* (Sec. 3.4). Pairwise Euclidean distance and surface-facing angles enter query-key dot products as exact SE(3)-invariant inner products, modulated by learned area and material gains. For novel views, a view decoder cross-attends camera rays to the scene tokens (Sec. 3.5). A lightweight point-splat rasterizer supplies primary hit depth, surface normals, and materials, allowing the cross-attention layers to focus on multi-bounce interreflections, soft contact shadows, and color bleeding.

Under matched parameter capacity (≈ 206 M parameters) and an identical 1.03M-scene dataset (Tab. 1), RADIANT reaches 25 dB tone-mapped PSNR in $5\times$ fewer optimizer steps than RenderFormer (16k vs. 80k) and finishes 600k steps at 32.2 dB (+3.3 dB over RenderFormer, +7.6 dB over RenderFormer-V2). Because sequence length scales with object count rather than polygon count (32N tokens vs. T triangles), a forward pass of RADIANT on a synthetic 500-object scene takes 0.83 s in a runtime-only benchmark, $8.5\times$ faster than RenderFormer with FlashAttention (Sec. 4.4); rendering quality is evaluated on scenes with up to 50 entities.

Our main contributions are:

- We introduce RADIANT, a mesh-free transformer that renders feed-forward global illumination directly from point inputs, decoupling transport complexity from polygon tessellation ($O(N^2)$ vs. $O(T^2)$ for N scene entities and T triangles, reducing median token counts from 3,205 to 320; Tab. 9).

- We formulate invariant form-factor attention, exactly factorizing selected geometric invariants motivated by radiative transfer into query-key dot products without pairwise bias tables, paired with a rasterizer-guided view decoder that resolves primary visibility without mesh connectivity.
- On our 1.03M-scene benchmark under matched training budgets, RADIANT reaches 25 dB in $5\times$ fewer optimizer steps than RenderFormer at the same global batch size, improves final fidelity by +3.3 dB, and runs a forward pass on 500-object scenes in 0.83 s in a runtime-only benchmark.

2 RELATED WORK

Per-scene neural rendering. Neural radiance fields (Mildenhall et al., 2020; Müller et al., 2022), 3D Gaussian splatting (Kerbl et al., 2023), and Neural Radiosity (Hadadan et al., 2021) optimize continuous representations for a single scene by minimizing photometric loss or the rendering equation’s residual (Kajiya, 1986). Although these methods avoid path tracing once trained, they require per-scene fitting and cannot generalize to novel geometric arrangements. In contrast, our goal is feed-forward synthesis: predicting global illumination across novel scenes in a single forward pass without test-time optimization.

Feed-forward rendering of explicit geometry. RenderFormer (Zeng et al., 2025) showed that transformers (Vaswani et al., 2017) can learn light transport over triangle meshes by tokenizing triangles with their material attributes. RenderFormer-V2 (Zeng et al., 2026) adds environment lighting and participating media with windowed attention, but still relies on triangle primitives. Concurrently, RenderFormer++ (Du et al., 2026) pools triangle tokens into object queries, yet still executes full self-attention over each object’s triangles before pooling, which incurs $O(\sum_i T_i^2)$ cost. RenderFormer++ is concurrent and has not released code; it requires triangle-based geometry and triangle-level attention, whereas RADIANT operates directly on point inputs via set abstraction (Qi et al., 2017b) without computing primitive attention. Other object-centric representations model scattering functions (Guo et al., 2020), object-oriented transfer (Zheng et al., 2023), superposed fields (Zheng et al., 2024), or light probes (Deng et al., 2025). In our framework, surface patches represent the physical scale of macroscopic light exchange directly.

Point-based and alternative representations for global illumination. Meshless representations have long decoupled illumination from surface tessellation (Lehtinen et al., 2008). Hermosilla et al. (2019) learn a direct-to-indirect illumination operator on 3D point clouds and propagate its features to image pixels, while Xu et al. (2026) map point clouds to continuous 3D shading queries for diffuse transport. Related approaches aggregate virtual point lights via pixel-light cross-attention (Ren et al., 2024) or predict screen-space shading from G-buffers (Nalbach et al., 2017). Here, RADIANT pools point inputs into 32 surface-patch tokens per object, models multi-bounce transfer in 3D self-attention, and decodes HDR radiance along camera rays guided by a point-splat G-buffer (Appendix K).

Rotary embeddings and attention biases. Spatial geometry and surface orientation determine radiative transfer. Rotary position embeddings (RoPE) (Su et al., 2024) encode relative offsets into query-key dot products, maintaining compatibility with fused attention kernels (Dao et al., 2022). PRoPE (Li et al., 2025) extends rotary embeddings to camera frustums for view synthesis, from which we adopt the CamRay parameterization. Rather than storing explicit $O(N^2)$ pairwise bias matrices, low-rank factorizations (Wu et al., 2025) decompose spatial biases into inner products. Invariant form-factor attention adopts this principle: Euclidean distance and surface-facing angles enter as exact inner products of per-token vectors, weighted by gains predicted from surface area and material attributes. We additionally incorporate feature-wise linear modulation (FiLM) (Perez et al., 2018) and register tokens (Darcet et al., 2024) to condition on shape statistics and prevent attention sinks.

3 METHOD

Problem setup. A scene is defined as a set of N objects $\{\mathcal{O}_i\}_{i=1}^N$, each specified by surface geometry with spatially varying material attributes (diffuse albedo, specular reflectance, roughness, and emission; emissive objects serve as light sources), together with a query camera with intrinsics

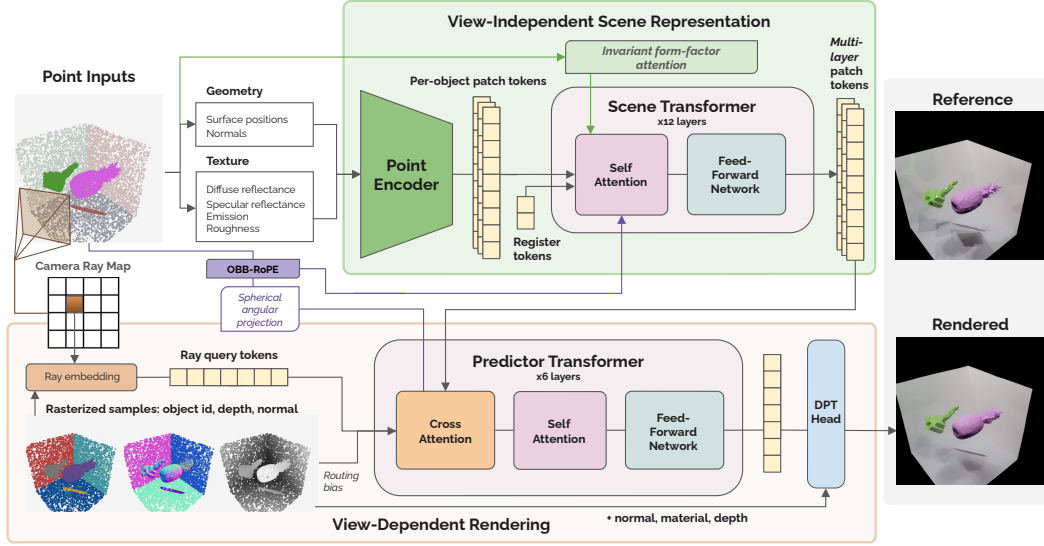


Figure 1: **Overview of RADIANT**, a single feed-forward network. View-independent part (computed once per scene): the point inputs are pooled into 32 patch tokens per object, and a scene transformer resolves 3D light transport with invariant form-factor attention. View-dependent part (computed per query viewpoint): the view decoder (predictor transformer), guided by rasterized geometry channels and spherical RoPE, cross-attends camera rays to scene tokens to decode the HDR image.

K and pose $[R|t]$. The goal is to synthesize, in a single forward pass for novel scenes, the HDR radiance image $L \in \mathbb{R}^{H \times W \times 3}$ that a reference path tracer would produce. We represent each object by $P=2,048$ surface point samples, each carrying a 16-dimensional attribute vector comprising position, normal, and ten material parameters. RADIANT is natively *mesh-free*. Polygonal meshes undergo uniform surface sampling, while raw point clouds with normals and materials are ingested directly without surface reconstruction. Our primary benchmarks evaluate mesh-derived samples, while Sec. 4.2 and Appendix J also evaluate simulated single-view partial point clouds.

3.1 ARCHITECTURE

Fig. 1 illustrates the network architecture. First, a point encoder maps each object’s point inputs to $M=32$ surface-patch tokens (Sec. 3.2). Global radiative transfer is modeled by a *scene transformer* (12 self-attention layers, width 1,024, 8 heads, feed-forward width 1,890, 97.0M parameters) with rotary encodings of object pose (Sec. 3.2) and invariant form-factor attention (Sec. 3.4). Next, for each viewpoint, the view decoder (or *predictor*, 6 layers with alternating cross- and self-attention, 96.1M parameters) renders the image by cross-attending camera ray tokens (constructed from Cam-Ray geometry and point-splat rasterization; Sec. 3.5) to the scene tokens. Finally, a dense prediction transformer (DPT) head (Ranftl et al., 2021) upsamples the decoded ray tokens from 4×4 pixel patches to full-resolution HDR radiance.

3.2 OBJECT TOKENIZATION

A PointNet-style encoder (Qi et al., 2017a) (10.0M parameters) embeds P point inputs per object in the object’s centered, scale-normalized coordinate frame. Using PointNet++ set abstraction (Qi et al., 2017b), farthest-point sampling (FPS) selects $M=32$ centroids per object with 128 nearest neighbors, where local max-pooling and linear projection yield 32 patch tokens of width 1,024 per object. Sixteen learnable register tokens (Darcet et al., 2024) absorb extraneous attention weight (-0.13 ± 0.01 dB when removed; Tab. 19). By construction, the scene transformer is equivariant to object permutations, ensuring that rendered imagery is invariant to object input ordering (Prop. 1, Appendix B).

To express relative 3D geometry, we inject object pose into attention queries and keys via an object-centric rotary embedding (*OBB-RoPE*). Each token is assigned a 12-dimensional oriented bounding box (OBB) descriptor $\mathbf{x}^{(i)} \in \mathbb{R}^{12}$, concatenating the patch centroid with the object’s three principal axes scaled by their standard deviations (computed in world space for the scene transformer and camera space for the predictor; Appendix F). The rotations of queries and keys compose into a function $R(\mathbf{x}^{(j)} - \mathbf{x}^{(i)})$ of the relative descriptor (Prop. 2).

3.3 RADIOSTY STATISTICS AND CONDITIONING

Each object carries eight radiosity statistics shared by its tokens: a log area proxy (1), mean diffuse albedo (3), mean emission (3), and scalar mean specular reflectance (1). Learned receiver and sender encodings $\delta \mathbf{q}_i$ and $\delta \mathbf{k}_j$ of these statistics are added to the queries and keys of the scene transformer before rotary encoding (Appendix F), and the same statistics determine the gains of invariant form-factor attention (Sec. 3.4). Two FiLM pathways (Perez et al., 2018) (2.1M parameters each) further condition each object’s tokens on rotation-invariant shape statistics of its point inputs and, in the view-dependent part, on its pose relative to the query camera.

3.4 INVARIANT FORM-FACTOR ATTENTION

Although rotary position embeddings capture relative positions, they do not directly expose the geometric invariants that govern radiative exchange, namely pairwise Euclidean distance r_{ij} and the angle θ_i between the receiver normal and the inter-patch displacement vector $\Delta_{ij} = \mathbf{c}_j - \mathbf{c}_i$, and the angle θ_j between the sender normal and $-\Delta_{ij}$. Patch centroids are centered at the scene mean and normalized by its root-mean-square radius (Prop. 4). We formulate an exact inner-product factorization that embeds these $O(3)$ -invariant geometric terms directly into query-key dot products. Each patch token carries its centroid $\mathbf{c}_i$ and unnormalized mean normal $\mathbf{m}_i$ (averaged over the 128 point inputs of its patch). We construct a 4-dimensional geometric invariant vector $\mathbf{G}_{ij} \in \mathbb{R}^4$ capturing the primary terms of radiative transfer (embedded via a 16-dimensional factored query-key representation):

$$\mathbf{G}_{ij} = \left(\underbrace{-\|\Delta_{ij}\|^2}_{\text{distance kernel}}, \underbrace{\mathbf{m}_i^\top \Delta_{ij}}_{\text{receiver-facing}}, \underbrace{-\mathbf{m}_j^\top \Delta_{ij}}_{\text{sender-facing}}, \underbrace{\mathbf{m}_i^\top \mathbf{m}_j}_{\text{mutual orientation}} \right). \quad (1)$$

Each of these four invariant scalar terms expands as an exact inner product between receiver and sender features. For example, the negative squared distance expands as $-\|\Delta_{ij}\|^2 = \langle [2\mathbf{c}_i, -\|\mathbf{c}_i\|^2, -1], [\mathbf{c}_j, 1, \|\mathbf{c}_j\|^2] \rangle$. Appending these factored features directly to the scene transformer queries and keys injects a learned form-factor prior without materializing a pairwise bias matrix. Across layers and attention heads, these terms are modulated by receiver and sender gains $a_t(i)$ and $b_t(j)$ predicted from radiosity statistics (Sec. 3.3). This contributes a learned scalar bias $\sum_{t=1}^4 a_t(i)b_t(j)G_{ij,t}$ to each attention logit without materializing the scene-space geometric bias matrix. Further, this bias is a material-dependent function of distance and orientation; when the product of the distance gains is positive, its squared-distance term contributes a Gaussian-shaped multiplicative factor to the unnormalized attention weights.

Theoretical properties and scope. These terms provide the exact $O(3)$ -invariant quadratic arguments of the differential radiative form factor $dF_{ij} = \cos \theta_i \cos \theta_j dA_j / (\pi r_{ij}^2)$ (Goral et al., 1984) (Prop. 4). Non-quadratic physical phenomena (specifically non-negative cosine clamping $[\cos \theta]_+$, inverse-square distance attenuation $1/r_{ij}^2$, and mutual visibility occlusion) are not explicitly represented by this factorization and are left to the learned transformer layers. All main experiments (Sec. 4.2) evaluate RADIANT with this formulation, and we isolate and measure its contribution in the ablation study (Sec. 4.3).

3.5 VIEW-DEPENDENT DECODING: RAY TOKENS, RASTERIZER CHANNELS, AND SPHERICAL CROSS-ATTENTION

Ray tokens. We tokenize camera rays by adopting an unnormalized variant of the CamRay parameterization (Li et al., 2025). Each pixel’s *CamRay*, obtained from $K^{-1}[u_{\text{pix}}, v_{\text{pix}}, 1]^\top$ and converted to the OpenGL camera frame, preserves the linearity of the image plane. Pixels are grouped into

4×4 patches. For each patch, the CamRay map and per-pixel channels described below are linearly projected and layer-normalized into a single ray token positioned at the patch’s CamRay centroid.

Rasterizer channels. To avoid burdening cross-attention with geometric ray casting, we separate primary surface visibility from global light transport. A depth-tested point-splat rasterization of the point inputs for the query camera (7×7 dilation) provides a mesh-free primary G-buffer with visible object identity, depth, normal, and material per pixel. The buffer does not explicitly encode incident illumination or shadows; cross-attention to the 3D scene tokens supplies global context for predicting direct and indirect illumination. Besides reflectance, the buffer’s material channels carry only the emission of directly visible emitters. The resulting log-depth, camera-space normal, and hit mask are patchified alongside the CamRay map, ensuring each ray token carries its primary hit geometry. Object identities enter cross-attention as a *routing bias*: the attention logit between a ray patch and an object’s tokens receives an additive bias proportional to the fraction of patch pixels rasterized to that object, scaled by a learned per-layer signed gate (initialized to zero). For output generation, rasterized normal, materials, and depth (14 channels) are added to full-resolution DPT features via a zero-initialized 1×1 convolution.

Spherical cross-attention RoPE. Cross-attention requires comparing 3D object positions against 2D image-plane coordinates under a unified relative encoding. We project both into a spherical coordinate frame consisting of two angles in $[-\pi, \pi]$ and a depth coordinate: an object at camera-space position (X, Y, Z) maps to $(\text{atan2}(X, -Z), \text{atan2}(Y, -Z), -Z)$, whereas a ray with normalized image-plane coordinates (u, v) , i.e., OpenGL camera-space direction proportional to $(u, v, -1)$, maps to $(\arctan u, \arctan v, \bar{d})$, where $\bar{d}$ denotes the mean rasterized hit depth of the patch (or 0 for missed rays). Under RoPE subtraction, the difference in the first two coordinates directly measures the angular offset between the ray and the object from the camera viewpoint, while the third coordinate provides explicit depth information for occlusion reasoning. These angular coordinates coincide exactly when a ray points toward the object’s centroid (Prop. 3).

3.6 TRAINING OBJECTIVE

The network is trained end-to-end. Following Zeng et al. (2025), the model predicts log radiance $\hat{y} = \log(1 + L)$. The loss $\mathcal{L}$ combines mean absolute error in log space with a perceptual term (Zhang et al., 2018) on tone-mapped images $\tau(L) = \text{clip}_{[0,1]}(\log_2(1 + L))$:

$$\mathcal{L} = \|\hat{y} - \log(1 + L^*)\|_1 + 0.05 \text{LPIPS}(\tau(\hat{L}), \tau(L^*)), \quad (2)$$

where the first term is averaged over all pixels and color channels, L^* denotes ground-truth radiance, and $\hat{L} = \exp(\min(\hat{y}, 10)) - 1$. We optimize using AdamW (Loshchilov & Hutter, 2019) with a peak learning rate of 5×10^{-5} , 8,000 warmup steps, cosine learning rate decay to 10^{-6} , and a global batch size of 24. All models share this loss, tone map, LPIPS backbone, and optimizer setup, except where RenderFormer-V2’s released training loop differs (Appendix C).

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Data. Following the RenderFormer protocol (Zeng et al., 2025), we construct procedural scenes with 1–6 Objaverse objects (Deitke et al., 2023) drawn from 18,085 meshes in a template room with randomized wall pruning, area lights (1–8 emitters per scene, mean 4.5; Tab. 9), and per-shading-group materials. Four views per scene are rendered with Blender Cycles at 512^2 resolution (512 samples per pixel, denoised via Intel Open Image Denoise; Appendix D). All models are trained from scratch on our 1.03M-scene corpus without external pre-rendered data. Apart from the frozen LPIPS backbone in the loss, the only pretrained components are the frozen encoders in RenderFormer-V2’s released input pipeline (Appendix F). The held-out evaluation split comprises 1,028 scenes (4,112 images) featuring novel object compositions, materials, emitters, and camera viewpoints. Each model ingests identical scenes in its native format (surface point samples for RADIANT, triangle meshes for baselines).

Matched-budget protocol. We benchmark three models of matched parameter capacity: RADIANT (205.9M), RenderFormer-Base (Zeng et al., 2025) (205.2M), and RenderFormer-V2 (207.6M;

Table 2: Held-out rendering quality under the matched-budget protocol (512^2 , 4,112 images). Each model is evaluated at its best checkpoint, selected over all evaluated checkpoints on the same split (Appendix C). Bold indicates best performance.

Model	PSNR $\uparrow$ (per-image / pooled)	LPIPS $\downarrow$	$\ell_1(\log)$ $\downarrow$
RADIANT (ours)	32.22 / 27.65	0.042	0.012
RenderFormer	28.95 / 25.80	0.077	0.019
RenderFormer-V2	24.60 / 23.05	0.117	0.046

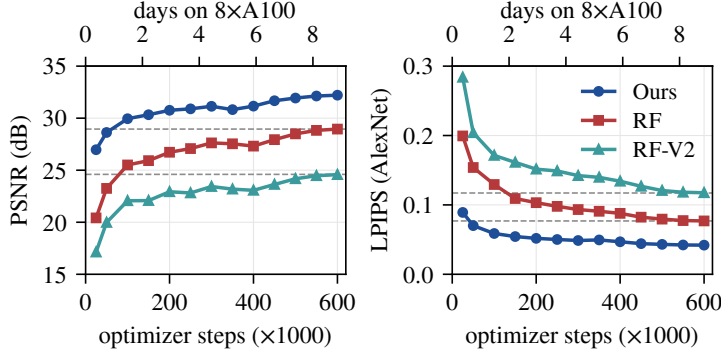


Figure 2: Learning curves of RADIANT (Ours), RenderFormer (RF), and RenderFormer-V2 (RF-V2) on 4,112 held-out images: per-image PSNR (left) and LPIPS (right). Top axis: $8 \times A100$ days at 1.28 s/step, the median step time of our model’s training run. Labels in RADIANT-equivalent time, not per-baseline wall-clock time. Dashed: best values of RF and RF-V2 (Tab. 2).

Appendix F), retrained under our unified protocol at 512^2 resolution on NVIDIA A100 GPUs (batch size 24, AdamW, bfloat16 mixed precision with FP32 master weights; Sec. 3.6). Training runs for 600k steps, starting with a 100k-step schedule and extended across successive cosine decays with periodic checkpoint evaluations (Sec. 4.2, Appendix C). Under the 256^2 protocol, seed-to-seed standard deviation is 0.01 dB, and paired bootstrap 95% confidence intervals confirm that our +3.3 dB and +7.6 dB performance margins are statistically significant (Appendix G).

Metrics. Models are evaluated on the 4,112 held-out images using three complementary metrics: (1) PSNR following Reinhard tone mapping (Reinhard et al., 2002) and sRGB conversion, computed both *per-image* and *pooled* across all pixels prior to logarithmic compression, (2) LPIPS (Zhang et al., 2018) with an AlexNet backbone, and (3) mean absolute error in log radiance $\ell_1(\log)$ (the regression term of the loss).

4.2 RENDERING QUALITY

RADIANT consistently outperforms both RenderFormer baselines across training, converging in substantially fewer optimizer steps. Learning curves (Fig. 2) and evaluation checkpoints (Tab. 8) show that by step 6k, RADIANT leads RenderFormer by +7.30 dB (23.45 vs. 16.15 dB) and cuts perceptual error by more than half (LPIPS 0.156 vs. 0.350). By step 16k, RADIANT reaches 25.91 dB, surpassing RenderFormer at step 50k (23.25 dB) and reaching 25 dB in five times fewer optimizer steps at the same global batch size (16k vs. 80k).

Upon completing the 600k-step schedule, RADIANT achieves 32.21 dB per-image PSNR (27.65 dB pooled), outperforming RenderFormer by +3.26 dB (+1.85 dB pooled) and RenderFormer-V2 by +7.63 dB, while reducing LPIPS by 45% and 64%, respectively (0.042 vs. 0.077 and 0.117; Tab. 8). Across training, the margin over RenderFormer remains substantial (+5.4 dB at 50k, +4.4 dB at 100k, +3.5 dB at 300k, and +3.3 dB at 600k). RenderFormer-V2, retrained with its authors’ released implementation under our matched budget, trails both models across all checkpoints. Across all evaluated checkpoints on the held-out split (Tab. 2; Appendix C), the best checkpoints are RADIANT at 596k (32.22 dB), RenderFormer at 600k (28.95 dB), and RenderFormer-V2 at 599k (24.60 dB).

RADIANT maintains a 3.1–3.5 dB margin over RenderFormer across all entity counts (Tab. 3; Appendix A). On an independent test set of 500 novel Objaverse assets (Tab. 4; Appendix I), RADIANT still leads RenderFormer and RenderFormer-V2 by +3.2 dB and +7.7 dB at training scene scales, and by +2.8 dB and +5.3 dB on crowded scenes with 25–50 entities.

Table 3: Held-out quality by scene complexity: the 4,112 evaluation images grouped by the number of entities the renderer sees (foreground objects, walls, and lights; the corpus spans 3 to 18). Per-image PSNR (dB) and LPIPS (AlexNet), with all three models at matched step 600k. The three models’ per-image scores are aligned by image index, which identifies the same procedural scene across models (Appendix D).

Entities per scene	Images	RADIANT (ours)		RenderFormer		RenderFormer-V2	
		PSNR $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	LPIPS $\downarrow$
≤ 7	796	32.35	0.037	29.22	0.064	24.46	0.104
8–10	1,376	32.15	0.041	29.00	0.075	24.52	0.116
11–13	1,336	32.44	0.043	29.11	0.079	24.80	0.118
≥ 14	604	31.64	0.051	28.12	0.094	24.35	0.136

Table 4: Unseen assets and larger scenes (Appendix I): per-image PSNR / LPIPS of the step-600k checkpoints; four views per scene. The held-out row repeats the 600k evaluation of Tab. 8. Bold: best per row.

Test set	Scenes	Entities	RADIANT (ours)	RenderFormer	RenderFormer-V2
Held-out split	1,028	3–18	32.21 / 0.042	28.95 / 0.077	24.57 / 0.117
A: unseen assets	250	3–17	32.59 / 0.041	29.36 / 0.073	24.90 / 0.110
B: 25–50 entities	280	25–50	25.32 / 0.095	22.50 / 0.171	20.06 / 0.227
C: object sizes of B	250	3–18	34.95 / 0.024	32.35 / 0.044	26.44 / 0.079

Table 5: Rendering from simulated single-view scans: the two held-out scenes of Fig. 4, RADIANT at step 600k (per-image PSNR, LPIPS, and ℓ_1 in log radiance; the evaluator of Sec. 4.1). *Original views*: the four views of each scene, which supervise the fine-tuning (eight images); *new views*: eight further views per scene from the training camera distribution, never used for fitting (16 images).

Point inputs	Fine-tuning	Original views			New views		
		PSNR $\uparrow$	LPIPS $\downarrow$	$\ell_1(\log) \downarrow$	PSNR $\uparrow$	LPIPS $\downarrow$	$\ell_1(\log) \downarrow$
Complete	none	33.87	0.034	0.004	34.24	0.037	0.004
Partial (single-view scan)	none	26.76	0.085	0.010	26.33	0.100	0.010
Partial (single-view scan)	400 steps	37.51	0.015	0.003	30.30	0.065	0.006
Complete	400 steps	39.02	0.009	0.002	33.58	0.038	0.005

On crowded held-out scenes (Fig. 3, step 600k), RADIANT resolves contact shadows, fine silhouettes, and color bleeding between nearby surfaces. In contrast, both RenderFormer baselines exhibit blurring and blocking artifacts across planar and curved geometry (detailed error breakdown in Appendix A).

The mesh-free formulation also handles incomplete point inputs without surface meshing. On novel viewpoints, RADIANT achieves 26.33 dB zero-shot from simulated single-view partial point clouds missing 52%–72% of each foreground object’s point inputs, compared to 34.24 dB from complete geometry (Tab. 5). Fine-tuning for 400 steps on four reference views raises novel-view PSNR to 30.30 dB, recovering half of the missing-geometry gap (Fig. 4, Appendix J).

4.3 ABLATIONS

Tab. 6 evaluates individual architectural components under a controlled 100k-step schedule at 256² resolution across 2,948 held-out images.

3D light transport modeling. Restricting scene attention to intra-object tokens degrades PSNR by 2.46 ± 0.66 dB across six random seeds and nearly doubles log-radiance error from 0.019 to 0.036 (Appendix G); the view decoder’s cross-attention to each object cannot replace global radiative exchange within the scene transformer. Removing invariant form-factor attention (Sec. 3.4) costs 0.52 ± 0.08 dB and raises LPIPS by 7%. A precision control confirms that float32 attention alone does not account for this gain: setting the invariant geometric terms to zero while retaining float32 attention degrades quality by 0.53 ± 0.02 dB (Tab. 19).

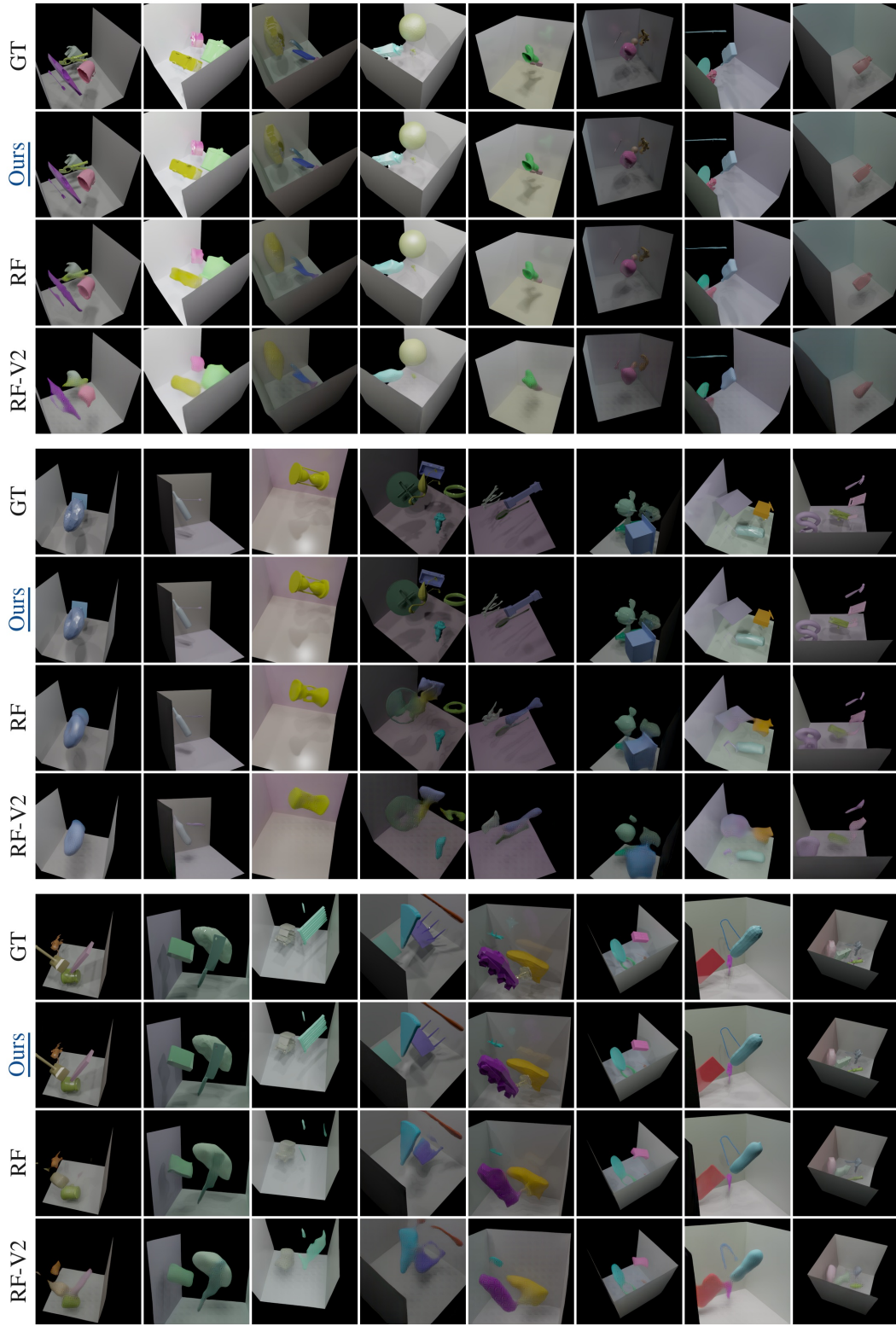


Figure 3: Qualitative comparisons on 24 crowded held-out scenes at matched step 600k. Rows in each group of eight scenes: ground truth (GT), RADIANT (Ours), RenderFormer (RF), and RenderFormer-V2 (RF-V2).

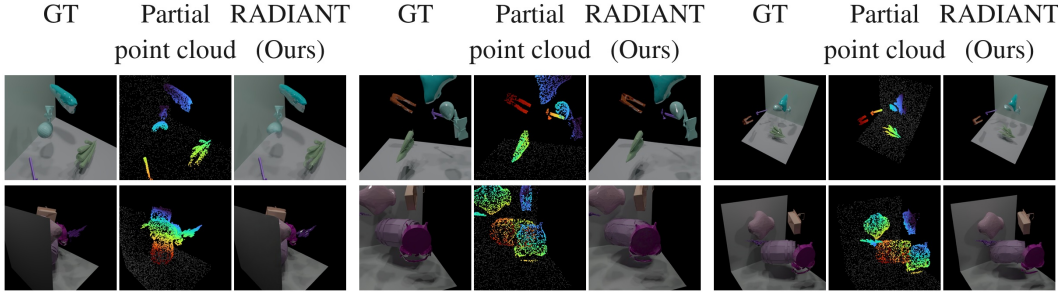


Figure 4: Rendering from partial point clouds (Appendix J) on two held-out scenes (rows), three novel views each, none used for fine-tuning. Each group of three columns: ground truth, the simulated single-view partial point cloud, and RADIANT after 400 fine-tuning steps on four reference views.

Table 6: Ablation of key mechanisms (256², 100k steps, mean over seeds 42–44; [†]seeds 42–47). PSNR per-image (mean $\pm$ sample standard deviation) / pooled; Δ : per-image change from the full model, paired within each seed ([†]seeds 45–47 against the full model’s three-seed mean). Extended ablations in Tab. 19.

Variant	PSNR $\uparrow$	LPIPS $\downarrow$	$\ell_1(\log) \downarrow$	Δ
Full model	29.96 $\pm$ 0.01 / 26.16	0.059	0.019	—
No invariant form-factor attention (§3.4)	29.43 $\pm$ 0.07 / 25.92	0.063	0.021	−0.52 $\pm$ 0.08
No inter-object scene attention (§3.1) [†]	27.50 $\pm$ 0.66 / 24.89	0.072	0.036	−2.46 $\pm$ 0.66
No rasterizer channels (hit geometry, routing bias)	26.84 $\pm$ 0.07 / 24.41	0.092	0.025	−3.12 $\pm$ 0.08

Explicit visibility grounding. Removing the rasterizer channels (hit geometry and routing bias) lowers PSNR by 3.12 ± 0.08 dB and increases LPIPS by 56% (Tab. 6), even though these channels carry only unshaded geometry (depth, normal, and hit mask). This shows providing explicit primary surface hits relieves cross-attention from having to reconstruct visibility. Conversely, on the model without invariant form-factor attention, decoding the rasterized G-buffer without 3D scene tokens lowers PSNR by 5.66 ± 0.06 dB (Tab. 20), demonstrating the importance of global 3D scene context beyond the view-local rasterized buffer.

Token efficiency and auxiliary modules. On the model without invariant form-factor attention, varying the patch token budget eightfold ($M \in \{8, 16, 32, 64\}$) alters PSNR by at most 0.04 dB (Tab. 15): allocating 16 patch tokens achieves 29.47 dB (matching 32 tokens at 29.43 dB) while reducing training step time by 17%, showing that compact patch sets reliably capture macroscopic transport. Conversely, pooling RenderFormer’s triangle tokens into 32 tokens per object lowers its PSNR by 0.38 ± 0.09 dB (Appendix G), indicating that architectural formulation rather than token budget accounts for RADIANT’s performance. Removing FiLM conditioning lowers PSNR by 0.15 ± 0.02 dB, while removing register tokens costs 0.13 ± 0.01 dB (Tab. 19).

4.4 INFERENCE EFFICIENCY AND SCALING WITH SCENE COMPLEXITY

Because RADIANT’s sequence length scales with object count ($32N$) rather than surface primitives, scene self-attention requires $O(N^2)$ operations regardless of geometric tessellation. On small scenes (up to 20 objects at 512^2), RenderFormer is faster because RADIANT’s runtime is dominated by decoding 16,384 ray tokens. Beyond 25 objects (35 with FlashAttention), RADIANT becomes faster. In a runtime-only benchmark with random weights on synthetic 500-object scenes, RADIANT runs $8.5\times$ faster than RenderFormer with FlashAttention and $22.2\times$ faster than standard PyTorch SDPA (benchmarks across tessellation levels, memory footprints, training costs, and throughput in Appendix E). In crowded regimes with 25–50 entities beyond the training distribution, rendering quality drops for all evaluated models (Tab. 4).

5 CONCLUSION

RADIANT shows that feed-forward global illumination can be learned directly from point inputs without mesh connectivity or per-scene optimization. By grouping point geometry into compact object-centric surface patches, the architecture decouples attention sequence length from polygon counts, while invariant form-factor attention factorizes geometric invariants of radiative transfer into query-key inner products without explicit pairwise bias tables. On a 1.03M-scene benchmark, RADIANT trains to 25 dB in one-fifth the steps of RenderFormer, achieves 32.2 dB at 600k steps, and, in a runtime-only benchmark with 500 objects, runs $8.5\times$ faster than RenderFormer.

Limitations and future work. Our formulation uses a uniform budget of 32 patch tokens per object. While effective for typical assets, this choice can over-allocate tokens to simple planar walls and under-sample geometrically complex objects. Allocating tokens adaptively based on surface area or curvature is left to future work. In addition, full self-attention over ray tokens in the view decoder limits scaling beyond 1024^2 resolution, which windowed or hierarchical ray attention could alleviate. Our point-splat rasterizer currently handles only direct surface intersections, leaving volumetric media such as smoke or fog to future work. Finally, high-frequency noise in input normals can propagate into rendered planar surfaces (Tab. 24). Pre-filtering normals with learned or bilateral smoothing could improve robustness on noisy real-world scans.

AI USE STATEMENT

Large language models (LLMs) were used to polish the manuscript prose and to assist with figure visualization.

ETHICS STATEMENT

This work uses only synthetic scenes assembled from publicly available Objaverse assets (Deitke et al., 2023) and involves no human subjects or personal data. All assets carry Creative Commons licenses. Of the 18,085 training meshes, 16,573 are CC BY, 739 CC BY-NC, 378 CC BY-SA, 354 CC BY-NC-SA, and 41 CC0; of the 500 meshes of the unseen-asset test set (Appendix I), 474 are CC BY, 14 CC BY-NC-SA, 8 CC BY-NC, and 4 CC BY-SA. The released scene manifest will list the license and author of every asset, and the non-commercial and share-alike terms of these licenses apply to the scenes that contain such assets.

REPRODUCIBILITY STATEMENT

The matched-budget protocol (Sec. 4.1) and Appendices C–F specify model architectures, parameter counts, dataset generation, loss functions, optimizer configurations, and evaluation schedules for all models. All models are trained under a matched configuration (Appendix C lists the details of RenderFormer-V2’s released training loop that differ) and scored with identical metric definitions. We will release our complete data-generation pipeline, scene corpus, training configurations, and evaluation code.

REFERENCES

- Jen-Hao Rick Chang, Wei-Yu Chen, Anurag Ranjan, Kwang Moo Yi, and Oncel Tuzel. Pointersect: Neural rendering with cloud-ray intersection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8359–8369, 2023. doi: 10.1109/CVPR52729.2023.00808.
- Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations (ICLR)*, 2024.
- Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3D objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- Yijie Deng, Lei Han, and Lu Fang. TransGI: Real-time dynamic global illumination with object-centric neural transfer model. *IEEE Transactions on Visualization and Computer Graphics*, 31(12):10457–10472, 2025. doi: 10.1109/TVCG.2025.3596146.
- Stavros Diolatzis, Julien Philip, and George Drettakis. Active exploration for neural global illumination of variable scenes. *ACM Transactions on Graphics*, 41(5), 2022. doi: 10.1145/3522735.
- Huangsheng Du, Haoran Zhu, Youcheng Cai, Jingyang Meng, and Ligang Liu. RenderFormer++: Scalable and physics-informed feed-forward neural rendering. *arXiv preprint arXiv:2606.30380*, 2026. v2, 7 August 2026.
- Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. In *Proceedings of SIGGRAPH*, 1984.
- Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503*, 2020.
- Saeed Hadadan, Shuhong Chen, and Matthias Zwicker. Neural radiosity. *ACM Transactions on Graphics*, 40(6), 2021.

- Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. In *Proceedings of SIGGRAPH*, pp. 197–206, 1991. doi: 10.1145/122718.122740.
- Pedro Hermosilla, Sebastian Maisch, Tobias Ritschel, and Timo Ropinski. Deep-learning the latent space of light transport. *Computer Graphics Forum*, 38(4):207–217, 2019. doi: 10.1111/cgf.13783.
- James T. Kajiya. The rendering equation. In *Proceedings of SIGGRAPH*, 1986.
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.
- Jaakko Lehtinen, Matthias Zwicker, Emmanuel Turquin, Janne Kontkanen, Frédo Durand, François X. Sillion, and Timo Aila. A meshless hierarchical representation for light transport. *ACM Transactions on Graphics*, 27(3), 2008. doi: 10.1145/1360612.1360636.
- Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2507.10496.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.
- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.
- Shaohua Mo, Chuankun Zheng, Zihao Lin, Dianbing Xi, Qi Ye, Rui Wang, Hujun Bao, and Yuchi Huo. Dual-band feature fusion for neural global illumination with multi-frequency reflections. In *ACM SIGGRAPH 2025 Conference Papers*, 2025. doi: 10.1145/3721238.3730733.
- Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. Real-time neural radiance caching for path tracing. *ACM Transactions on Graphics*, 40(4), 2021. doi: 10.1145/3450626.3459812.
- Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4), 2022.
- Oliver Nalbach, Elena Arabadzhiyska, Dushyant Mehta, Hans-Peter Seidel, and Tobias Ritschel. Deep shading: Convolutional neural networks for screen space shading. *Computer Graphics Forum*, 36(4):65–78, 2017. doi: 10.1111/cgf.13225.
- Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, 2018.
- Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017a.
- Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017b. arXiv:1706.02413.
- René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.
- Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. *ACM Transactions on Graphics*, 21(3):267–276, 2002.
- Haocheng Ren, Yuchi Huo, Yifan Peng, Hongtao Sheng, Weidong Xue, Hongxiang Huang, Jingzhen Lan, Rui Wang, and Hujun Bao. LightFormer: Light-oriented global neural rendering in dynamic scene. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 43(4), 2024. doi: 10.1145/3658229.
- Paul Sanzenbacher, Lars Mescheder, and Andreas Geiger. Learning neural light transport. *arXiv preprint arXiv:2006.03427*, 2020.

- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- Rui Su, Honghao Dong, Haojie Jin, Yisong Chen, Guoping Wang, and Sheng Li. Vertex features for neural global illumination. In *SIGGRAPH Asia 2025 Conference Papers*, 2025. doi: 10.1145/3757377.3763826.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- Soledad Villar, David W. Hogg, Kate Storey-Fisher, Weichi Yao, and Ben Blum-Smith. Scalars are universal: Equivariant machine learning, structured like classical physics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- Hermann Weyl. *The Classical Groups: Their Invariants and Representations*. Princeton University Press, 2nd edition, 1946.
- Haixu Wu, Minghao Guo, Yuezhou Ma, Yuanxu Sun, Jianmin Wang, Wojciech Matusik, and Ming-sheng Long. FlashBias: Fast computation of attention with bias. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.
- Bing Xu, Mukund Varma T, Cheng Wang, Tzu-Mao Li, Lifan Wu, Bartlomiej Wronski, Ravi Ramamoorthi, and Marco Salvi. A generalizable light transport 3D embedding for global illumination. In *ACM SIGGRAPH 2026 Conference Papers*, 2026. doi: 10.1145/3799902.3811095.
- Chong Zeng, Yue Dong, Pieter Peers, Hongzhi Wu, and Xin Tong. RenderFormer: Transformer-based neural rendering of triangle meshes with global illumination. In *ACM SIGGRAPH 2025 Conference Papers*, 2025. doi: 10.1145/3721238.3730595.
- Chong Zeng, Yue Dong, Pieter Peers, Lvmin Zhang, and Maneesh Agrawala. RenderFormer-V2: Neural rendering with heterogeneous scene primitives. In *European Conference on Computer Vision (ECCV)*, 2026. arXiv:2609.05738.
- Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- Chuankun Zheng, Yuchi Huo, Shaohua Mo, Zhihua Zhong, Zhizhen Wu, Wei Hua, Rui Wang, and Hujun Bao. NeLT: Object-oriented neural light transfer. *ACM Transactions on Graphics*, 42(5), 2023. doi: 10.1145/3596491.
- Chuankun Zheng, Yuchi Huo, Hongxiang Huang, Hongtao Sheng, Junrong Huang, Rui Tang, Hao Zhu, Rui Wang, and Hujun Bao. Neural global illumination via superposed deformable feature fields. In *SIGGRAPH Asia 2024 Conference Papers*, 2024. doi: 10.1145/3680528.3687680.
- Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. Surface splatting. In *Proceedings of SIGGRAPH*, pp. 371–378, 2001. doi: 10.1145/383259.383300.

A ERROR ANALYSIS AND SCENE COMPLEXITY

Remaining errors. In Fig. 3, the errors of RADIANT concentrate on thin structures, which the rasterizer’s 7×7 dilation renders slightly thicker than in the reference, and on the boundary between an object and its cast shadow, where the hit channels and the routing bias change discontinuously.

Performance breakdown by scene complexity. The gap measured over the whole held-out split is consistent across subsets of scenes. Tab. 3 splits the same 4,112 images by the number of entities the renderer sees, from the sparsest bucket (at most 7 entities, 796 images) to the most crowded. At matched step 600k, RADIANT’s lead over RenderFormer is 3.1–3.5 dB in every bucket, and its lead over RenderFormer-V2 is 7.3–7.9 dB, confirming that the performance hierarchy of Tab. 2 is not an artifact of scene complexity. Absolute PSNR varies by at most 1.1 dB with the entity count for all three models, which is expected: the corpus holds one to six foreground objects in a template room, so the buckets differ in how much of the image is occupied rather than in the difficulty of the transport. The perceptual metric does separate them: from the sparsest to the most crowded bucket LPIPS rises by 37% for RADIANT, 47% for RenderFormer, and 31% for RenderFormer-V2; the fixed budget of surface-patch tokens per object does not make crowded scenes disproportionately harder. This is a statement about rendering quality only, as the inference cost of RADIANT and RenderFormer at large object counts is measured separately in Tab. 11.

B PROPERTIES OF THE GEOMETRIC MECHANISMS

B.1 STATEMENTS

Proposition 1 (Object-permutation equivariance). *The scene transformer is equivariant to permutations of the object set, and the predictor’s output is invariant to them: for any permutation π of $\{1, \dots, N\}$, applying π to the objects permutes the scene-transformer token blocks by π and, provided the rasterization selects the same surface at every pixel under relabeling (no depth ties between different objects), leaves the rendered image unchanged.*

Proposition 2 (Relative-pose dependence). *For the rotary block, the attention logit between patch tokens i and j depends on $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ only through their difference: $\tilde{\mathbf{q}}_i^\top \tilde{\mathbf{k}}_j = \mathbf{q}_i^\top R(\mathbf{x}^{(j)} - \mathbf{x}^{(i)}) \mathbf{k}_j$ for a block-diagonal rotation R determined by the frequencies, for fixed pre-rotation vectors $\mathbf{q}_i, \mathbf{k}_j$. In particular, a common translation of all centroids leaves the rotary-block logits between patch tokens unchanged.*

Proposition 3 (Angular alignment). *For a ray with normalized image-plane coordinates (u, v) and an object at camera-space position (X, Y, Z) with $Z < 0$, the first two spherical coordinates coincide exactly when the ray passes through the object’s centroid, and their difference is the pair of view angles between the ray and the centroid direction. The angular coordinates are therefore invariant to reparameterizations of the intrinsics and resolution that leave the physical rays unchanged.*

Proposition 4 (Invariant form-factor attention). *Let $\mathbf{x}_i = (\mathbf{c}_i - \bar{\mathbf{c}})/\rho$ be the token centroids normalized per scene ($\bar{\mathbf{c}}$ the mean and ρ the root-mean-square radius of the scene’s patch centroids), let $\mathbf{G}_{ij}$ be as in Eq. 1 with $\Delta_{ij} = \mathbf{x}_j - \mathbf{x}_i$, and let $\ell_{ij} = \sum_t a_t(i) b_t(j) G_{ij,t}$ be the geometric term of the logit.*

1. *Each $G_{ij,t}$ equals $\langle \phi_t(i), \psi_t(j) \rangle$ for per-token features ϕ_t, ψ_t of total width 16.*
2. *ℓ_{ij} is unchanged when every centroid is mapped to $R\mathbf{c} + \mathbf{t}$ and every normal to $R\mathbf{m}$, for any $R \in O(3)$ and $\mathbf{t} \in \mathbb{R}^3$. $\mathbf{G}_{ij}$ is moreover unchanged under a uniform scaling $\mathbf{c} \mapsto s\mathbf{c}$, $s > 0$, and ℓ permutes with the tokens.*
3. *Every $O(3)$ -invariant polynomial in $(\Delta_{ij}, \mathbf{m}_i, \mathbf{m}_j)$ that is homogeneous of degree two is a linear combination of the four entries of $\mathbf{G}_{ij}$, $\|\mathbf{m}_i\|^2$, and $\|\mathbf{m}_j\|^2$.*

B.2 PROOFS

Proof of Proposition 1. The encoder of Sec. 3.2 is applied independently to each object, so permuting the objects permutes its outputs. The OBB descriptor $\mathbf{x}^{(i)}$, the radiosity encodings $\delta \mathbf{q}_i, \delta \mathbf{k}_i$, and the FiLM statistics are each functions of object i alone, so they permute with the tokens; the

register tokens' position is the mean patch centroid, which is permutation-invariant. Self-attention with per-token positional terms is permutation-equivariant, hence so is the scene transformer. In the predictor, ray tokens attend to the set of patch tokens through a softmax over the set, which is permutation-invariant. The rasterizer channels depend on the union of the objects' surfaces and, under the hypothesis that no two objects tie for the nearest depth at a pixel, select the same surface after relabeling. The routing bias attaches to each object's tokens by identity. The rendered image is therefore invariant. $\square$

Proof of Proposition 2. Write the rotary block as $\tilde{\mathbf{q}}_i = R(\mathbf{x}^{(i)})\mathbf{q}_i$ and $\tilde{\mathbf{k}}_j = R(\mathbf{x}^{(j)})\mathbf{k}_j$, where $R(\mathbf{x})$ is block-diagonal with 2×2 rotations by angles $\omega_f x_d$ for coordinate d and frequency ω_f . Then $\tilde{\mathbf{q}}_i^\top \tilde{\mathbf{k}}_j = \mathbf{q}_i^\top R(\mathbf{x}^{(i)})^\top R(\mathbf{x}^{(j)})\mathbf{k}_j = \mathbf{q}_i^\top R(\mathbf{x}^{(j)} - \mathbf{x}^{(i)})\mathbf{k}_j$ because planar rotations commute and compose additively. A common translation $\mathbf{t}$ of all centroids adds $\mathbf{t}$ to the first three coordinates of every patch descriptor and leaves their differences, hence the rotary-block logits between patch tokens, unchanged. The content vectors are translation-invariant because the encoder sees centered samples and the FiLM and radiosity statistics are translation-invariant. $\square$

Proof of Proposition 3. A ray with normalized image-plane coordinates (u, v) has camera-space direction proportional to $(u, v, -1)$, so its horizontal and vertical view angles are $\arctan u$ and $\arctan v$. An object centroid at (X, Y, Z) with $Z < 0$ has view angles $\text{atan2}(X, -Z)$ and $\text{atan2}(Y, -Z)$. The two pairs coincide exactly when $(X, Y)/(-Z) = (u, v)$, i.e., when the ray passes through the centroid, and their differences are the angular offsets between the ray and the centroid direction. Both quantities are defined from camera-space geometry and the corresponding unnormalized camera-space ray, so a change of focal length or resolution that leaves the physical ray unchanged leaves the coordinates unchanged. $\square$

Proof of Proposition 4. 1. By expansion,

$$\begin{aligned} -\|\mathbf{x}_j - \mathbf{x}_i\|^2 &= \langle [2\mathbf{x}_i, -\|\mathbf{x}_i\|^2, -1], [\mathbf{x}_j, 1, \|\mathbf{x}_j\|^2] \rangle, \\ \mathbf{m}_i^\top (\mathbf{x}_j - \mathbf{x}_i) &= \langle [\mathbf{m}_i, -\mathbf{m}_i^\top \mathbf{x}_i], [\mathbf{x}_j, 1] \rangle, \\ -\mathbf{m}_j^\top (\mathbf{x}_j - \mathbf{x}_i) &= \langle [\mathbf{x}_i, 1], [\mathbf{m}_j, -\mathbf{m}_j^\top \mathbf{x}_j] \rangle, \\ \mathbf{m}_i^\top \mathbf{m}_j &= \langle \mathbf{m}_i, \mathbf{m}_j \rangle, \end{aligned}$$

of widths $5 + 4 + 4 + 3 = 16$; scaling the query features of term t by $a_t(i)$ and the key features by $b_t(j)$ makes ℓ_{ij} their inner product.

2. Under $\mathbf{c} \mapsto R\mathbf{c} + \mathbf{t}$ the mean maps to $R\bar{\mathbf{c}} + \mathbf{t}$ and ρ is unchanged, so $\mathbf{x} \mapsto R\mathbf{x}$; with $\mathbf{m} \mapsto R\mathbf{m}$, every entry of $\mathbf{G}_{ij}$ is an inner product of Δ_{ij} , $\mathbf{m}_i$, $\mathbf{m}_j$ and is preserved because $R^\top R = I$. Under $\mathbf{c} \mapsto s\mathbf{c}$, ρ scales by s , so $\mathbf{x}$ is unchanged, and the normals do not change. The gains are functions of per-object statistics (log area, albedo, emission, specular reflectance) that are invariant to rigid motions, though the log area is not invariant to scaling. Hence ℓ_{ij} is invariant to rigid motions. Every quantity is a function of its own token and of permutation-invariant scene statistics $(\bar{\mathbf{c}}, \rho)$, so ℓ permutes with the tokens.

3. By the first fundamental theorem of invariant theory for $O(3)$ (Weyl, 1946), used for equivariant networks by Villar et al. (2021), every $O(3)$ -invariant polynomial in vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ is a polynomial in their inner products $\langle \mathbf{v}_a, \mathbf{v}_b \rangle$. Those homogeneous of degree two are linear combinations of the inner products themselves, which for $(\Delta_{ij}, \mathbf{m}_i, \mathbf{m}_j)$ are the four entries of $\mathbf{G}_{ij}$ up to sign, $\|\mathbf{m}_i\|^2$, and $\|\mathbf{m}_j\|^2$. The last two depend on one token alone: a query-only term cancels in the softmax, and a key-only term is a per-key offset. $\square$

C TRAINING DETAILS

Tab. 7 summarizes the training configuration and hyperparameters in our experiments. All models are trained from scratch at 512^2 on NVIDIA A100 GPUs with a global batch size of 24 (without data augmentation). We optimize using AdamW (Loshchilov & Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay of 0.01 applied to matrix weights, and gradient clipping at a global norm of 1.0. To train RenderFormer, we use its released model code, Base configuration, and training initialization within the training loop shared with RADIANT. RenderFormer-V2 is trained also with

its released training loop, with the data loader replaced to read our scenes and the LPIPS backbone set to AlexNet—this loop applies the weight decay to all parameters, warms up from zero, anneals to 5×10^{-7} , and casts the targets to bfloat16. Both baselines keep their view-dependent stage in TF32, as in their authors’ code.

The learning rate warms up linearly over 8,000 steps to a peak of 5×10^{-5} , followed by cosine annealing to a minimum of 10^{-6} at 100k steps. Longer runs continue from the 100k checkpoint on the tail of a 200k-step cosine schedule, from the 200k checkpoint on that of a 300k-step one, and from the 300k checkpoint on that of a 600k-step one (same warm-up, peak, and minimum; model and optimizer state carried over), so the learning rate rises at each continuation, for example to 2.6×10^{-5} at step 300k; the main comparison is at step 600k. Training uses bfloat16 mixed precision with FP32 master weights.

Every saved checkpoint is evaluated on the full held-out split of 4,112 images with identical metric definitions (one evaluator for RADIANT and RenderFormer, a port of it to RenderFormer-V2’s released code); the best checkpoint of each model (Tab. 2) is the one with the highest per-image PSNR on this split, selected identically for the three models. Choosing each model’s checkpoint on a random half of the held-out scenes and scoring it on the other half (1,000 halvings) selects the 600k checkpoint in every halving for both RenderFormer generations and in 99.9% of the choices for RADIANT (otherwise its 575k checkpoint) among the regular evaluations from 100k steps on, and the best checkpoints of Tab. 2 exceed these choices by at most 0.02 dB.

Table 7: Training hyperparameters of the main comparison. The ablations (Sec. 4.3) use 256^2 images, 100k steps, seeds 42–44 (42–47 without inter-object attention), and 2,948 held-out images.

Hyperparameter	Value	Hyperparameter	Value
Hardware	NVIDIA A100 GPUs	Gradient clipping	Global norm 1.0
Global batch size	24	Mixed precision	bf16 autocast, fp32 master weights
Optimizer	AdamW, $(\beta_1, \beta_2) = (0.9, 0.999)$	Loss function	MAE in $\log(1+L) + 0.05$ LPIPS
Peak learning rate	5×10^{-5}	LPIPS backbone	AlexNet
Warm-up	8,000 linear steps	Image resolution	512×512
Learning rate decay	Cosine to 10^{-6} at 100k/200k/300k/600k	Data augmentation	None (1 view per sample)
Weight decay	0.01	Evaluation split	Held-out (4,112 images)
Random seed	42 (initialization and data order; RenderFormer-V2’s released loop sets it after building the model)	Total training steps	100k, continued to 600k on longer schedules; matched comparison at 600k

D DATA STATISTICS

Tab. 9 summarizes the corpus. Scenes follow the RenderFormer protocol: one to six foreground objects from a pool of 18,085 curated Objaverse meshes, placed in a template room whose walls are randomly pruned, with randomized area lights, materials assigned per shading group, and meshes simplified by quadric edge collapse. Each scene is path traced in Blender Cycles from four cameras at 512^2 with 512 samples per pixel, and a second corpus generated by the same procedure is rendered at 256^2 with 256 samples per pixel. Both corpora use the Cycles defaults of Blender 4.2: adaptive sampling (noise threshold 0.01), an indirect-light clamp of 10, and denoising with Intel Open Image Denoise. Every entity the renderer sees (foreground objects, walls, and lights) is one object of the model, which is why the entity count exceeds the foreground count. The held-out split is the last 0.1% of scenes in a fixed order, and we verify that the point-sampled and triangle representations carry identical scene sets in identical order and that the split boundary falls on the same scene. The held-out scenes share the curated pool of 18,085 Objaverse meshes with the training set, but they are distinct scenes: their compositions, arrangements, material assignments, emitters, and camera viewpoints are sampled independently of the training scenes. Per-scene statistics are measured on a random sample of 40,000 training scenes (40 data files of 1,000 scenes).

Table 8: Evaluation points of Fig. 2 (step 25k and every 50k) and the early steps quoted in Sec. 4.2 (6k, 16k, 80k): per-image PSNR (dB) and LPIPS (AlexNet) on the 4,112 held-out images. —: not evaluated. Bold: best value of each run.

Step	RADIANT (ours)		RenderFormer		RenderFormer-V2	
	PSNR $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	LPIPS $\downarrow$
6k	23.45	0.156	16.15	0.350	—	—
16k	25.91	0.103	19.02	0.244	16.39	0.326
25k	26.97	0.089	20.42	0.199	17.13	0.284
50k	28.63	0.070	23.25	0.154	19.98	0.204
80k	29.70	0.061	25.06	0.133	21.75	0.175
100k	29.95	0.059	25.50	0.129	22.06	0.171
150k	30.33	0.054	25.92	0.109	22.08	0.161
200k	30.76	0.052	26.72	0.103	22.93	0.152
250k	30.90	0.050	27.08	0.098	22.81	0.149
300k	31.14	0.049	27.62	0.093	23.43	0.142
350k	30.82	0.050	27.54	0.091	23.18	0.140
400k	31.16	0.047	27.31	0.088	23.07	0.134
450k	31.67	0.044	27.94	0.082	23.63	0.127
500k	31.93	0.043	28.49	0.079	24.18	0.121
550k	32.13	0.042	28.83	0.077	24.47	0.118
600k	32.21	0.042	28.95	0.077	24.57	0.117

Table 9: Corpus statistics. Entity, light, and triangle statistics are from a random sample of 40,000 training scenes (40 data files of 1,000 scenes); p5/p50/p95 are percentiles.

Scenes (after exclusions)	1,027,596
Images per scene / total images	4 / 4,110,384
Resolutions / samples per pixel	512 ² / 512 and 256 ² / 256
Training / held-out scenes	1,026,568 / 1,028 (4,112 images)
Foreground objects per scene	1–6 (from 18,085 meshes)
Entities per scene (objects, walls, lights)	mean 10.1; p5/p50/p95 = 5/10/15; max 18
Emissive entities per scene	mean 4.5; range 1–8
Triangles per scene (simplified)	mean 3,196; p5/p50/p95 = 1,155/3,205/5,630; cap 8,192
Point inputs per object (RADIANT)	2,048, with normals and 10 material parameters
Excluded scenes	322 (154 cap, 119 archive, 48 timeout, 1 remesh)

E RUNTIME ANALYSIS

Complexity. With N objects, $M=32$ patch tokens per object, and 16 register tokens, the scene transformer attends over $S = 32N + 16$ tokens at cost $O(S^2)$ per layer; the predictor attends $Q = HW/16$ ray tokens over the scene tokens at cost $O(QS)$ and over themselves at cost $O(Q^2)$ ($Q = 16,384$ at 512^2); the rasterization is one splat pass over the NP point inputs ($P = 2,048$) with a 7×7 dilation; the DPT head is linear in HW . None of these terms depends on the triangle count of the input meshes. A triangle-level model attends over T tokens at cost $O(T^2)$ in its first stage.

Protocol. Tabs. 11 and 14: the two 206M models of Tab. 2 with random weights, synthetic scenes (2,048 point inputs per object sampled on spheres, with analytic normals, for RADIANT; random small triangles for RenderFormer; surface sampling is not timed), bf16 autocast, batch size 1, one NVIDIA H100 GPU (80 GB; PyTorch 2.7.1, CUDA 12.6), five warm-up and up to twenty timed forward passes per configuration (median reported). Tab. 16: the same models evaluated on eight-object scenes (512 triangles per object for RenderFormer, 2,048 points per object for RADIANT), with five warm-up and twenty timed passes, excluding host-to-device transfers. Fig. 5 plots the latencies of Tab. 11 against the number of objects, together with the latency of a training step. Complete benchmarking scripts and profiling configurations are included in the supplementary materials.

Breakdown of the forward pass. Tab. 10 splits the forward pass into its components. The view-dependent part dominates RADIANT at 10 and 50 objects (68% and 58% of the time) and grows only from 50 to 75 ms up to 200 objects, while the point encoder and the scene transformer, whose attention runs in float32 (Sec. 3.4), grow with the number of objects (7.3 to 48.0 ms and 9.0 to 104.0 ms). RenderFormer’s triangle transformer grows from 19 ms to 1.20 s with FlashAttention

over the same range and takes 94% of the forward pass at 200 objects (96% with PyTorch attention).

Table 10: Inference time per component (ms, median of 20 forwards, CUDA events around the top-level modules) of the two 206M models at 512^2 on one H100 with random weights, batch size 1, and bf16, for 10, 50, and 200 objects (2,048 point inputs or 512 triangles per object). Encoder: point encoder of RADIANT or triangle attribute encodings of RenderFormer; transport: scene transformer or triangle transformer; view decoder + DPT: view-dependent transformer and DPT head; other: rasterization, conditioning, and the remaining operations. RenderFormer with FlashAttention (flash) and, at 200 objects, PyTorch attention (sdpa). Profiled separately from Tab. 11.

Component	RADIANT			RenderFormer			
	10	50	200	10 (flash)	50 (flash)	200 (flash)	200 (sdpa)
Encoder	7.3	16.0	48.0	0.3	0.9	3.0	3.0
Transport	9.0	16.0	104.0	19.2	116.1	1,203.0	2,959.9
View decoder + DPT	50.2	55.7	75.4	15.6	26.8	71.4	101.5
Other	7.5	7.7	20.6	1.2	1.4	2.6	2.9
Total	74.0	95.4	248.0	36.3	145.3	1,280.0	3,067.3

Table 11: Single-forward inference time (ms, median) and peak memory (MiB) vs. number of objects (512^2 , batch size 1, H100), from 2,048 point inputs per object for RADIANT and 512 triangles per object for RenderFormer (surface sampling not timed). RenderFormer runs with PyTorch SDPA or FlashAttention (its memory is that of the SDPA run); speedups divide its time by RADIANT’s.

Objects (triangles)	RADIANT		RenderFormer			Speedup	
	time	memory	sdpa	flash	memory	sdpa	flash
1 (512)	70.4	4,255	19.4	30.2	1,631	0.28×	0.43×
10 (5,120)	73.3	4,299	33.6	35.9	1,644	0.46×	0.49×
50 (25,600)	95.7	4,480	260.8	145.4	1,954	2.7×	1.5×
100 (51,200)	136.8	5,863	848.1	398.5	2,859	6.2×	2.9×
200 (102,400)	248.5	10,898	3,065.2	1,288.9	4,669	12.3×	5.2×
500 (256,000)	826.9	25,996	18,326.0	6,990.0	10,101	22.2×	8.5×

Crossover. Measured between 10 and 50 objects with the settings of Tab. 11, RenderFormer with SDPA is faster at 512^2 up to 20 objects (70.9 versus 77.2 ms at 20), and RADIANT is faster from 25 objects on; against FlashAttention the crossover lies between 30 and 35 objects. At 128^2 and 256^2 , RADIANT is faster from 15 objects on (at 128^2 from 10 against FlashAttention).

Tessellation. Tab. 12 keeps ten objects and varies only their tessellation. RenderFormer’s time grows with the triangle count, while RADIANT’s input (2,048 points per object) and time do not change: from 2,048 triangles per object on, RADIANT is faster (1.5–2.5×), and at 32,768 triangles per object by 155× (FlashAttention) to 412× (SDPA).

Table 12: Single-forward inference time (ms, median) for ten objects as their tessellation grows (512^2 , batch size 1, H100, settings of Tab. 11). RADIANT reads 2,048 point inputs per object regardless of tessellation. Speedups divide RenderFormer’s time by RADIANT’s.

Triangles per object (total)	RADIANT	RF sdpa	RF flash	Speedup (sdpa)	Speedup (flash)
512 (5,120)	72.7	33.5	35.5	0.46×	0.49×
2,048 (20,480)	72.7	182.0	109.8	2.5×	1.5×
8,192 (81,920)	72.7	2,030	872.5	27.9×	12.0×
32,768 (327,680)	72.7	29,965	11,309	411.9×	155.5×

Cost at the scale of the training scenes. On the small scenes of the corpus the balance reverses (Tab. 13, one A100): a single-view forward of RADIANT takes 130 ms against RenderFormer’s 42–49 ms, and for a training step it runs 1.09 s against 0.72–0.91 s. With 5× fewer steps taken to 25 dB, RADIANT still reaches it in 4.8 h of training against 15.9–20.3 h.

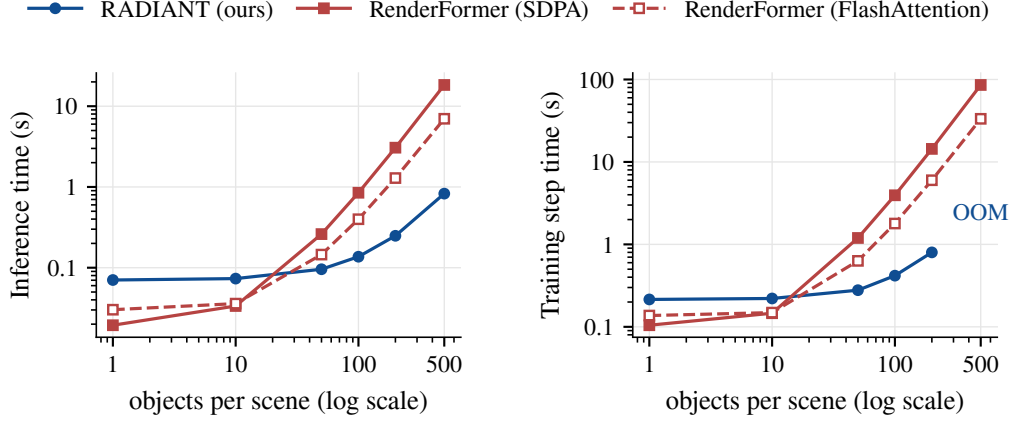


Figure 5: Inference forward latency (left) and training step latency (right) of the two 206M models versus the number of objects (512^2 , batch size 1, bf16 autocast, one H100 80 GB GPU; 512 triangles per object for RenderFormer, 2,048 point inputs per object for RADIANT). RenderFormer is evaluated with PyTorch SDPA (solid) and FlashAttention (dashed). Training step includes forward pass, loss computation, backward pass, and AdamW update. In complex scenes with 50–500 objects, RADIANT provides up to $22.2\times$ faster inference than RenderFormer with PyTorch SDPA and $8.5\times$ faster than RenderFormer with FlashAttention. Training steps of RADIANT are $4.3\text{--}18.0\times$ faster than RenderFormer’s with SDPA and $2.3\text{--}7.5\times$ faster than with FlashAttention at 50–200 objects; at 500 objects RADIANT exceeds the 80 GB of GPU memory in training, whereas RenderFormer needs 32.7 GiB. Inference latencies are tabulated in Tab. 11.

Table 13: Cost at the scale of the training scenes on one A100 80 GB GPU (corpus scenes). Inference: median time per image on the held-out scenes, batch size 1, bf16 autocast as in Tab. 11; RenderFormer with SDPA / FlashAttention. Training step: median time per optimizer step with three training scenes from the corpus (the per-GPU share of the global batch of 24) on a single A100 GPU, each model in its training configuration: RADIANT under bf16 autocast, RenderFormer with / without activation recomputation. Time to 25 dB: steps to 25 dB (Tab. 1) multiplied by per-step latency on an 8-GPU node without communication overhead.

Model	Inference (ms)	Training step (s)	Steps to 25 dB	Time to 25 dB (h)
RADIANT	130	1.09	16k	4.8
RenderFormer	42 / 49	0.91 / 0.72	80k	15.9–20.3

Batched throughput. On small scenes the batched regime (Tab. 16) favors RADIANT at 128^2 and 256^2 from batch size 8 on, where object-centric tokens allow efficient parallel processing; at 512^2 , ray tokens dominate per-image overhead.

Multi-view amortization. Both models compute a view-independent scene representation once and replay only the view-dependent part per view. Tab. 17 times the two parts separately: the split reproduces the output of the unmodified forward pass to within bf16 rounding, and the sum of the two parts matches one end-to-end forward within 0.9%. RADIANT’s view-independent part is the cheaper one from 50 objects on (38 versus 118 ms under FlashAttention, 38 versus 227 ms under the shared backend), and the gap grows with the object count as in Tab. 11. Its view-dependent part, however, costs 52–58 ms at 512^2 , 2.1–3.3 times RenderFormer’s view transformer, because it decodes 16,384 ray tokens through the DPT head. At 512^2 the amortized per-view cost therefore favors RenderFormer on these 8- and 50-object scenes (16.0 and 29.6 versus 52.6 and 58.5 ms per view at 64 views). At 256^2 the view-dependent parts are comparable (13–21 ms) and the view-independent part decides: the two models are within a millisecond per view at eight objects and RADIANT is ahead at fifty (16.1 versus 22.4 ms at 64 views).

Table 14: Single-forward inference time (ms, median) at 128^2 and 256^2 for the settings of Tab. 11. Speedups divide RenderFormer’s time by RADIANT’s.

Objects	RADIANT	RenderFormer		Speedup	
		sdpa	flash	sdpa	flash
128 ²					
1	28.6	17.4	28.9	0.61×	1.01×
10	30.3	25.4	33.1	0.84×	1.09×
50	47.6	245.7	137.8	5.2×	2.9×
100	80.8	823.6	384.7	10.2×	4.8×
200	180.2	3,022.4	1,267.7	16.8×	7.0×
500	722.7	18,236.4	6,959.2	25.2×	9.6×
256 ²					
1	31.9	17.4	28.9	0.55×	0.91×
10	34.0	26.5	33.2	0.78×	0.98×
50	52.7	246.5	138.3	4.7×	2.6×
100	88.4	823.8	386.7	9.3×	4.4×
200	191.2	3,023.4	1,269.5	15.8×	6.6×
500	740.6	18,260.0	6,964.9	24.7×	9.4×

Table 15: Token budget per object, on the ablation subset and held-out split of Tab. 6 under its 100k-step schedule, on the model without invariant form-factor attention (Sec. 3.4), mean over seeds 42–44: PSNR per-image (mean $\pm$ sample standard deviation) / pooled, LPIPS, and ℓ_1 in log radiance. Step time is the relative training step latency normalized to $M=32$.

Patch tokens per object M	PSNR $\uparrow$	LPIPS $\downarrow$	$\ell_1(\log) \downarrow$	Step time
8	29.45 ± 0.02 / 25.93	0.062	0.021	$0.79\times$
16	29.47 ± 0.02 / 25.93	0.062	0.021	$0.83\times$
32 (default)	29.43 ± 0.07 / 25.92	0.063	0.021	$1.00\times$
64	29.44 ± 0.06 / 25.92	0.063	0.021	$1.09\times$

Table 16: Batched throughput (ms per image, median) of the two 206M models on eight-object scenes (one H100 80 GB GPU, bf16, five warm-up and twenty timed forward passes, host-to-device transfers excluded; 512 triangles per object for RenderFormer, 2,048 points per object for RADIANT).

Batch size	128^2			256^2			512^2	
	1	8	32	1	8	32	1	8
RenderFormer (sdpa)	21.5	13.7	12.8	22.5	15.2	14.4	29.2	22.3
RenderFormer (flash)	30.6	12.4	10.7	30.4	13.5	12.1	33.0	19.4
RADIANT (sdpa, ours)	30.3	7.1	5.5	33.1	13.2	11.9	73.2	51.3

F ARCHITECTURE

Tab. 18 lists the modules of RADIANT with their parameter counts (205,942,972 in total). The scene transformer uses QK-LayerNorm, rotary frequencies $1, 5^{1/3}, 5^{2/3}, 5$ over the 12 descriptor coordinates (96 rotated channels, 32 pass-through, rotate-half convention), 16 register tokens positioned at the mean patch centroid, and the receiver/sender radiosity encodings of Sec. 3.3. The predictor fuses the last three scene-transformer layers into its context, uses rotary frequencies $1, 3^{1/3}, 3^{2/3}, 3$ over the spherical ray/object coordinates, adds the routing bias with one learned gate per layer, and decodes the ray tokens of the 4×4 pixel patches with a DPT head (256 features, output channels 128/256/512/1,024) that also receives the 14 rasterized surface channels through a zero-initialized 1×1 convolution.

RenderFormer (205.2M): the released Base configuration, 12 view-independent and 6 view-dependent layers, width 768, 6 heads, feed-forward 3,072, DPT decoder, 8×8 ray patches.

Table 17: Multi-view amortization of the two 206M models (one H100, bf16, batch size 1, random weights, synthetic scenes of 512 triangles per object, FlashAttention for both, three warm-up and ten timed iterations, medians). t_{scene} : view-independent part (RADIANT: encoder and scene transformer; RenderFormer: triangle-token transformer); t_{view} : view-dependent part (RADIANT: rasterization, ray tokens, predictor and DPT head; RenderFormer: camera-space triangle positions, ray generation, view transformer and DPT head). The per-view cost of V views is $(t_{\text{scene}} + V t_{\text{view}})/V$. RADIANT’s scene transformer computes its float32 attention through PyTorch SDPA under both backends. Under the shared sdpa backend t_{scene} is 20.0/37.7 ms for RADIANT and 15.2/227 ms for RenderFormer at 8/50 objects; t_{view} changes by less than 1 ms for RADIANT and becomes 14.8/34.7 ms at 512^2 for RenderFormer.

Resolution	Model	Objects	t_{scene} (ms)	t_{view} (ms)	ms/view, $V=1$	$V=16$	$V=64$
512^2	RADIANT	8	19.5	52.3	71.8	53.5	52.6
	RenderFormer	8	17.7	15.7	33.4	16.8	16.0
	RADIANT	50	37.7	57.9	95.7	60.3	58.5
	RenderFormer	50	117.5	27.8	145.3	35.1	29.6
256^2	RADIANT	8	19.4	13.9	33.3	15.1	14.2
	RenderFormer	8	17.5	13.3	30.9	14.4	13.6
	RADIANT	50	37.7	15.5	53.2	17.9	16.1
	RenderFormer	50	117.6	20.5	138.1	27.9	22.4

RenderFormer-V2 (207.6M): the released 200M configuration, which orders tokens along a Hilbert curve and restricts attention among them to a sliding window of ± 256 tokens (FlexAttention) with attention sinks (16 register tokens and one summary token per 64 geometry tokens), uses a Swin-style view transformer, and encodes 32×32 material patches into $4 \times 4 \times 80$ VAE latents projected to the token width; frozen pretrained encoders of its released input pipeline (a material-to-latent mapper and an image VAE) produce these latents and that of the empty environment map.

Table 18: Modules of RADIANT and their parameter counts, totaling 205,942,972 parameters as counted from a trained checkpoint. Widths and layer counts are those of Sec. 3.1.

Module	Configuration	Parameters
Point encoder	16-D inputs, hidden 512/1,024/1,024, backbone 2,048, GroupNorm; 32 FPS patch tokens of 128 neighbors	9,985,024
Scene transformer	12 layers, width 1,024, 8 heads of 128, FFN 1,890, QK-LayerNorm, OBB-RoPE	96,962,716
Register tokens	16	16,384
Receiver/sender encoders	two MLPs $8 \rightarrow 256 \rightarrow 1,024$	530,944
Invariant-attention gains	two MLPs $8 \rightarrow 64 \rightarrow 384$ (Sec. 3.4)	51,072
Object FiLM	intrinsic-complexity statistics $\rightarrow$ scale/shift	2,112,512
Ray encoder	CamRay map + 5 hit channels per pixel, 4×4 patches	135,168
Predictor	6 cross+self-attention layers, width 1,024, FFN 1,496, spherical RoPE, routing gates, camera FiLM (2,108,416), DPT head	96,149,152
Total		205,942,972

Implementation details omitted from Sec. 3. OBB-RoPE: each patch token i carries $\mathbf{x}^{(i)} = [\mathbf{c}^{(i)} \parallel \mathbf{e}_1^{(i)} \parallel \mathbf{e}_2^{(i)} \parallel \mathbf{e}_3^{(i)}] \in \mathbb{R}^{12}$, its patch centroid and the principal axes of its parent object’s samples (covariance eigendecomposition) scaled by the standard deviations along them, shared by the object’s tokens; each coordinate is mapped through $F=4$ logarithmically spaced frequencies to phases $\mathbf{a}^{(i)}$, and queries and keys are split into a 96-dimensional rotary block and a 32-dimensional pass-through block, $\tilde{\mathbf{q}} = [\mathbf{q}_r \odot \cos \mathbf{a}^{(i)} + \text{rot}(\mathbf{q}_r) \odot \sin \mathbf{a}^{(i)} \parallel \mathbf{q}_p]$ with the rotate-half pairing $\text{rot}(\cdot)$.

Register tokens are positioned at the masked mean of the scene’s patch centroids (with a small numerical epsilon). Eigendecompositions and camera transforms of the descriptor are evaluated in single precision; in the predictor, centroids receive the full rigid camera transform and axes the

rotation only. The rotate-half convention pairs each channel of the first half of the rotary block with its partner in the second half, with a sign flip.

The receiver/sender MLPs of the radiosity encodings (Sec. 3.3) are $8 \rightarrow 256 \rightarrow 1,024$. The intrinsic-complexity FiLM reads covariance spectra, radial moments, and convexity of the object’s samples. CamRay assumes square images with a centered principal point; the ray-token projection is followed by a LayerNorm. Patches without rasterized hits carry a depth coordinate of zero in the spherical frame.

Invariant form-factor attention (Sec. 3.4). The receiver and sender gains are two MLPs $8 \rightarrow 64 \rightarrow 12 \cdot 8 \cdot 4$ over the eight radiosity statistics of Sec. 3.3, one gain per layer, head, and term; the last layer of the sender MLP is initialized at zero, so the invariant terms start at exactly zero. The 16 features per head are appended to the queries and keys after the rotary rotation, the values are padded with zeros, and the attention of these layers is computed in float32 with the softmax scale of the 128-dimensional heads. Centroids are centered at the mean of the scene’s patch centroids and divided by their root-mean-square radius (Prop. 4), which keeps the float32 expansion well conditioned; each mean normal averages, without normalization, the normals of the 128 samples nearest to the patch centroid, the neighborhood of the point encoder, so a patch whose normals cancel carries a short vector rather than an arbitrary direction; register tokens carry a zero normal. The gradients of the gain MLPs are clipped separately from the rest of the model. The extension adds 0.05M parameters (the two gain MLPs, $8 \rightarrow 64 \rightarrow 384$; Tab. 18).

G SEED REPLICATES OF THE ABLATIONS

Table 19: Ablation rows (256^2 , 100k steps, mean over seeds 42–44; † seeds 42–47) with the metrics of Tab. 6. PSNR per-image (mean $\pm$ sample standard deviation) / pooled; Δ : per-image change from the full model, paired within each seed († seeds 45–47 against the full model’s three-seed mean). Below the rule: two controls of invariant form-factor attention (Sec. 3.4).

Variant	PSNR $\uparrow$	LPIPS $\downarrow$	$\ell_1(\log) \downarrow$	Δ
Full model	29.96 $\pm$ 0.01 / 26.16	0.059	0.019	—
No invariant form-factor attention (§3.4)	29.43 $\pm$ 0.07 / 25.92	0.063	0.021	-0.52 ± 0.08
No inter-object scene attention (§3.1) †	27.50 $\pm$ 0.66 / 24.89	0.072	0.036	-2.46 ± 0.66
No FiLM conditioning (§3.3)	29.81 $\pm$ 0.02 / 26.09	0.060	0.019	-0.15 ± 0.02
No register tokens	29.82 $\pm$ 0.02 / 26.10	0.060	0.019	-0.13 ± 0.01
No rasterizer channels (hit geometry, routing bias)	26.84 $\pm$ 0.07 / 24.41	0.092	0.025	-3.12 ± 0.08
Invariant terms frozen at zero (float32 attention kept)	29.43 $\pm$ 0.03 / 25.92	0.063	0.021	-0.53 ± 0.02
No OBB-RoPE (invariant attention kept)	29.87 $\pm$ 0.04 / 26.12	0.059	0.019	-0.08 ± 0.04

Rasterizer controls. Tab. 20 separates the rasterizer channels from the scene representation under the 100k-step schedule of Tab. 19, over seeds 42–44, on the model without invariant form-factor attention (row “No invariant form-factor attention” of Tab. 19). Removing the rasterized splat into the output head leaves PSNR unchanged ($+0.01 \pm 0.07$ dB). Removing the scene tokens from the view decoder while keeping every rasterizer channel and the splat costs 5.66 ± 0.06 dB and triples the log-radiance error (0.062 against 0.021), more than removing the rasterizer channels (3.12 ± 0.05 dB). Removing the rasterizer channels and the splat together costs 3.09 ± 0.20 dB. The rasterizer channels identify the surface each ray meets, and the scene tokens provide the global 3D scene context needed to shade it: neither suffices alone.

Tab. 19 reports the mean of seeds 42, 43, and 44 (42–47 without inter-object attention) under the same 100k-step schedule, and Tab. 21 lists every seed. Each replicate changes both the initialization and the data order; all runs are scored by the same evaluator on the same 2,948 held-out images. The full model’s seed-to-seed standard deviation is 0.01 dB. Paired within seeds, removing the rasterizer channels costs 3.12 ± 0.08 dB and removing invariant form-factor attention 0.52 ± 0.08 dB; removing FiLM costs 0.15 ± 0.02 dB and removing the register tokens costs 0.13 ± 0.01 dB, with the same sign in every seed. Without inter-object attention the loss depends on the seed, from 1.41 to 3.03 dB, so this row reports seeds 42–47; seeds 45–47 have no full-model run of their own and are compared

Table 20: Rasterizer controls under the 100k-step schedule of Tab. 19 (mean over seeds 42–44), on the model without invariant form-factor attention. Columns indicate whether the view decoder receives the scene tokens, the rasterizer channels (hit channels and routing bias), and the rasterized splat into the output head. PSNR per-image (mean $\pm$ sample standard deviation) / pooled; Δ : per-image change from the reference row, paired within each seed.

Variant	Scene tokens	Rasterizer channels	Splat	PSNR $\uparrow$	LPIPS $\downarrow$	$\ell_1(\log)$ $\downarrow$	Δ
Reference (no invariant attention)	✓	✓	✓	29.43 $\pm$ 0.07 / 25.92	0.063	0.021	—
No rasterized splat	✓	✓	✗	29.44 $\pm$ 0.04 / 25.92	0.062	0.021	+0.01 $\pm$ 0.07
No rasterizer channels	✓	✗	✓	26.31 $\pm$ 0.09 / 24.08	0.099	0.027	−3.12 $\pm$ 0.05
No rasterizer channels or splat	✓	✗	✗	26.34 $\pm$ 0.13 / 24.01	0.094	0.027	−3.09 $\pm$ 0.20
No scene tokens	✗	✓	✓	23.77 $\pm$ 0.01 / 21.44	0.109	0.062	−5.66 $\pm$ 0.06

with its three-seed mean. The rasterized splat into the output head (Sec. 3.5) is kept in every row. The result is thus carried by the rasterizer channels, the attention between objects, and invariant form-factor attention, not by the auxiliary conditioning.

Table 21: Per-image PSNR (dB) of every row of Tab. 19 at 100k steps (the schedule of Tab. 6) for seeds 42, 43, and 44 (42–47 without inter-object attention), with the mean and sample standard deviation. Δ is paired within each seed, against the full model; seeds 45–47 against its three-seed mean. Below the rule: two controls of invariant form-factor attention (Sec. 3.4).

Variant	Seed						Mean $\pm$ SD	Δ
	42	43	44	45	46	47		
Full model	29.97	29.95	29.96	—	—	—	29.96 $\pm$ 0.01	—
No invariant form-factor attention	29.36	29.49	29.45	—	—	—	29.43 $\pm$ 0.07	−0.52 $\pm$ 0.08
No inter-object scene attention [†]	28.55	26.95	26.93	27.69	26.98	27.89	27.50 $\pm$ 0.66	−2.46 $\pm$ 0.66
No FiLM conditioning	29.79	29.81	29.82	—	—	—	29.81 $\pm$ 0.02	−0.15 $\pm$ 0.02
No register tokens	29.84	29.81	29.82	—	—	—	29.82 $\pm$ 0.02	−0.13 $\pm$ 0.01
No rasterizer channels	26.76	26.88	26.88	—	—	—	26.84 $\pm$ 0.07	−3.12 $\pm$ 0.08
Invariant terms frozen at zero	29.43	29.41	29.46	—	—	—	29.43 $\pm$ 0.03	−0.53 $\pm$ 0.02
No OBB-RoPE	29.83	29.88	29.90	—	—	—	29.87 $\pm$ 0.04	−0.08 $\pm$ 0.04

Inter-object attention. Restricting the self-attention of the scene transformer to the tokens of each object (the register tokens form a block of their own, so nothing relays between objects; the view decoder still cross-attends to every object’s tokens, emitters included) lowers per-image PSNR by 2.46 ± 0.66 dB over seeds 42–47 (pooled 1.27 ± 0.36 dB; LPIPS +23%, $\ell_1(\log)$ +91%). The loss depends on the seed: 1.41, 2.07, and 2.27 dB for seeds 42, 47, and 45, and 2.98 to 3.03 dB for seeds 43, 44, and 46. The drop is present at every scene size and largest in the sparsest scenes: 2.87, 2.47, 2.35, and 2.23 dB for at most 7, 8–10, 11–13, and at least 14 entities (seeds 42–44, paired per image within each seed), where the walls and floor lit by the emitters fill more of the image.

Invariant form-factor attention. Trained from scratch under the protocol of Tab. 6, the full model reaches 29.96 ± 0.01 dB per-image PSNR against 29.43 ± 0.07 dB without the invariant channels of Sec. 3.4, a paired difference of 0.52 ± 0.08 dB over seeds 42–44 (pooled 26.16 against 25.92 dB, LPIPS 0.059 against 0.063, $\ell_1(\log)$ 0.019 against 0.021; three-seed means). Two controls isolate the mechanism. The invariant layers compute attention in float32, the rest of the model in bf16; the full model with the gains frozen at their initialization, so that the invariant terms are exactly zero, reaches 29.43 ± 0.03 dB over seeds 42–44, 0.53 ± 0.02 dB below the full model and level with the model without the invariant channels (-0.00 ± 0.08 dB paired): precision does not account for the gain. Without OBB-RoPE (no rotary encoding in the scene transformer, invariant channels kept) the

model reaches 29.87 ± 0.04 dB, 0.08 ± 0.04 dB below the full model and 0.44 ± 0.04 dB above the model without the invariant channels (paired), so the invariants carry more than the rotary encoding.

Controls. On the model without invariant form-factor attention, varying the token budget per object over $M \in \{8, 16, 32, 64\}$ (Tab. 15) moves the three-seed mean by at most 0.04 dB from the default $M=32$, so the fixed budget is not the binding constraint at this scale.

Pooling RenderFormer’s triangles per object. RADIANT represents every object by a fixed budget of 32 patch tokens. To test whether this budget alone explains the difference from RenderFormer, we give RenderFormer the same budget: before global transport, a learned pooler maps the triangle tokens of each object to 32 tokens (learned queries at 32 farthest-point anchors per object and two layers of RenderFormer’s own transformer decoder; +11.6% parameters), and both of its transformers then run on the pooled tokens instead of the triangles. With the authors’ released training loop and the 100k-step schedule of Tab. 19, pooling lowers RenderFormer’s per-image PSNR from 23.56 ± 0.21 to 23.19 ± 0.15 dB, by 0.38 ± 0.09 dB paired within seeds (Tab. 22), while RADIANT reaches 29.96 ± 0.01 dB under the same schedule.

The difference between the two architectures is therefore not a matter of the token budget per object. Within RADIANT, the ablations above attribute the quality to the rasterizer channels and the attention between objects. In the pooled model the view-dependent stage no longer attends to individual triangles, which may account for part of the loss. RADIANT provides the primary visibility through its rasterizer channels instead.

Table 22: RenderFormer with the triangle tokens of each object pooled into 32 tokens before global transport, trained with the authors’ released training loop under the 100k-step schedule of Tab. 19 (mean over seeds 42–44). PSNR per-image (mean $\pm$ sample standard deviation) / pooled; Δ : per-image change from RenderFormer without pooling, paired within each seed. The last row is the full model of Tab. 19.

Model	PSNR $\uparrow$	LPIPS $\downarrow$	$\ell_1(\log) \downarrow$	Δ
RenderFormer	$23.56 \pm 0.21 / 21.84$	0.166	0.041	—
RenderFormer, 32 pooled tokens per object	$23.19 \pm 0.15 / 21.59$	0.165	0.041	-0.38 ± 0.09
RADIANT (full model)	$29.96 \pm 0.01 / 26.16$	0.059	0.019	—

Uncertainty of the main comparison. Paired per-image PSNR differences at matched step 600k, with 95% confidence intervals from a scene-level bootstrap (the four views of a held-out scene are resampled together; 10,000 resamples): RADIANT leads RenderFormer by +3.26 dB ($[+3.12, +3.39]$) and RenderFormer-V2 by +7.63 dB ($[+7.45, +7.81]$), and it is better on 90.1% and 98.1% of the 4,112 held-out images (92.1% and 98.2% of the 1,028 scenes). Each model is trained once (seed 42), so these intervals cover the sampling of held-out scenes, not training variance; under the ablation protocol, the sample standard deviation of per-image PSNR across seeds 42–44 is 0.01 dB for RADIANT and 0.21 dB for RenderFormer (Tabs. 6 and 22).

H DESIGN TRADE-OFFS

Tab. 23 places RADIANT among feed-forward renderers of explicit scenes.

Token budget per object. The fixed budget is what makes the cost independent of mesh resolution and what the mechanisms of Sec. 3 compensate for. The ablation shows that the compensation is uneven: the rasterizer channels, which tell the decoder where each pixel lands, recover three decibels (Tab. 6), whereas removing OBB-RoPE from the scene transformer costs 0.08 ± 0.04 dB (Tab. 19); and Tab. 15 shows, on the model without invariant form-factor attention, that the budget itself can be varied eightfold within 0.04 dB ($29.45 / 29.47 / 29.43 / 29.44$ dB for $M=8/16/32/64$), with 16 tokens matching 32 at 17% lower step time. Objects whose appearance varies strongly across their surface remain the hard case, and a variable token budget per object is the natural extension.

Attention backend. The two models favor different attention kernels: RenderFormer’s dense triangle attention gains $2.6\times$ from FlashAttention at 500 objects (Tab. 11), whereas RADIANT’s routed cross-attention materializes a ray-by-token bias and therefore runs through PyTorch’s scaled-dot-

Table 23: Feed-forward renderers of explicit scenes: geometric input, what the sequence length is proportional to (T triangles, N objects, M learned queries per object), full global illumination without per-scene optimization (GI), and a per-scene representation reused across views (Amortized).

Method	Input	Tokens	GI	Amortized
Per-scene fields (Mildenhall et al., 2020; Kerbl et al., 2023)	images	—	per scene	—
RenderFormer (Zeng et al., 2025)	triangles	$O(T)$	✓	✓
RenderFormer-V2 (Zeng et al., 2026)	triangles, media, environment maps	$O(T)$, windowed	✓	✓
RenderFormer++ (Du et al., 2026)	triangles	$O(T_i)$ per object, then $O(NM)$	✓	✓
RADIANT (ours)	point inputs (from meshes or point clouds)	$O(N)$, 32 per object	✓	✓

product path, as does the float32 attention of its scene transformer (Sec. 3.4). Efficiency comparisons report both the shared backend and RenderFormer’s fastest one.

Disentangling 3D transport, explicit visibility, and decoder capacity. To determine whether RADIANT’s gains stem from structured 3D light transport modeling, explicit surface visibility from the point rasterizer, or view decoder capacity (16k versus 4k ray tokens, both decoded by a DPT head), we address three questions:

- **3D light transport vs. explicit visibility:** While the point rasterizer provides local surface grounding (depth, normals, materials, instance mask), it contains no indirect illumination or shadows. In the control without scene tokens (Tab. 20, on the model without invariant form-factor attention), the view decoder receives all rasterizer channels and ray tokens but no 3D scene tokens, and PSNR drops by 5.66 ± 0.06 dB. Conversely, holding the rasterizer and decoder fixed, restricting the 3D scene transformer to attention within objects lowers PSNR by 2.46 ± 0.66 dB and removing invariant form-factor attention by 0.52 ± 0.08 dB (Tab. 6). Thus, with the rasterizer and decoder fixed, variants of the 3D scene transformer span 2.5 dB (27.50 to 29.96 dB).
- **Decoder capacity and ray token density:** At 512^2 , RADIANT uses 16,384 ray tokens (4×4 patches) and RenderFormer uses 4,096 tokens (8×8 patches), both with a DPT decoder, at matched total parameter counts (≈ 206 M). The control without the rasterized splat (Tab. 20, same model) removes the per-pixel feature injection into the DPT head and leaves PSNR unchanged, so late convolutional upsampling does not bypass cross-attention transport. Training RADIANT (without invariant form-factor attention) with RenderFormer’s 8×8 ray patches under the ablation protocol (Tab. 6) lowers per-image PSNR by 0.60 ± 0.09 dB, paired over seeds 42–44 (28.83 ± 0.03 against 29.43 ± 0.07 dB): the denser ray tokens contribute, but less than the attention between objects in the scene transformer, whose removal costs this model 1.59 ± 0.25 dB over seeds 42–44 (the full model 2.46 ± 0.66 dB, Tab. 6).
- **Decoupling visibility from transport:** Integrating rasterized point channels directly into triangle transformers would fundamentally alter their input representation and cross-attention formulation away from mesh primitives. The controls of Tab. 20 isolate the role of rasterized guidance within the point-based paradigm, demonstrating that global 3D scene context remains essential.

I UNSEEN ASSETS AND LARGER SCENES

The held-out split shares the training meshes and the training range of scene sizes (Appendix D). To test generalization beyond both, we build a separate test set with the same generator and 500 Objaverse objects that never occur in training. The curation criteria admit no mesh beyond the 18,085 of the training pool, so the new objects are Objaverse-LVIS objects that meet every curation criterion except the Sketchfab-category filter; none matches a training mesh by identifier, file hash, or geometry. Three subsets are rendered and converted exactly like the corpus (four views per scene

at 512²): *A*, 250 scenes from the unchanged generator (3–17 entities, mean 9.7, against 10.3 on the held-out split); *B*, 280 scenes with 25–50 entities, whose n objects are scaled by $(3.10/n)^{1/3}$ so that they fill the room as in an average training scene; and *C*, 250 scenes with the training object count and the object sizes of *B*, which separates object size from entity count. Every *B* scene exceeds the 8,192-triangle limit of the RenderFormer data (10,153–44,200 triangles after the same simplification, median 27,766); RenderFormer and RenderFormer-V2 receive these scenes with the limit raised for this evaluation only, so all three models run beyond their training range. Tab. 4 reports the step-600k checkpoints with the metrics of Sec. 4.1; confidence intervals come from a scene-level bootstrap.

On unseen assets at the training scene size (*A*), each model scores +0.32 to +0.41 dB relative to its held-out result, so the new assets cost no accuracy. With 25–50 entities (*B*), all three models lose accuracy ($B - A$: −7.3, −6.9, and −4.8 dB for RADIANT, RenderFormer, and RenderFormer-V2), and RADIANT still leads RenderFormer by +2.81 dB (95% CI [+2.63, +3.00]) and RenderFormer-V2 by +5.26 dB ([+5.03, +5.49]). Smaller objects alone (*C*) are easier for all three models ($C - A$: +2.4, +3.0, and +1.5 dB), so the drop on *B* comes from the number of entities rather than from object size. RADIANT leads both baselines on every subset: by +3.23 and +7.69 dB on *A* and by +2.61 and +8.52 dB on *C*.

J IMPERFECT GEOMETRY AND ROBUSTNESS

Raw geometry acquired by physical depth sensors is frequently incomplete due to occlusions. To assess how RADIANT behaves under unobserved surface regions, we evaluate two held-out scenes (five foreground objects each) where each object retains only the point inputs visible from a single simulated depth camera at the first view: points that are camera-facing, within the viewing frustum, and pass a z-buffer test against entity point splats (radius 2 pixels at 512²). Across objects, 28%–48% of the point inputs survive this visibility test. Resampling these to 2,048 points per object preserves the standard input budget while covering only the observed portions of each object; room surfaces and emitters remain complete.

Zero-shot vs. fine-tuning protocol. We evaluate RADIANT at the 600k-step checkpoint in two regimes (Tab. 5), on the four original views of each scene and on eight new views per scene, path traced like the corpus from cameras drawn from the training camera distribution and never used for fitting. (1) *Zero-shot inference*: without fine-tuning, the partial scans reach 26.33 dB on the new views (26.76 dB on the original views), against 34.24 dB from complete geometry; unscanned surfaces render as voids. (2) *Per-scene adaptation*: fine-tuning on the four original views for 400 steps (batch size 2, AdamW, learning rate 10^{-5} , using the loss in Sec. 3.6) brings the partial scans to 37.51 dB on those views, 1.51 dB below the same fine-tuning from complete geometry (39.02 dB), but to 30.30 dB on the new views, and the fine-tuning from complete geometry does not carry over to the new views (33.58 dB against 34.24 dB without fine-tuning). The gains on the fitted views therefore mostly reflect fitting those views; on views that were not fitted, fine-tuning recovers 50% of the 7.91 dB lost to the missing surfaces. Fig. 4 shows three of the new views per scene.

Zero-shot rendering from corrupted inputs. Tab. 24 evaluates the 600k-step checkpoint without fine-tuning across various geometric corruptions on the first 1,028 held-out images. When corruptions are restricted to foreground objects (room and emitters clean), reducing point density to a quarter (512 samples per entity) or adding Gaussian position noise with $\sigma = 0.01$ of the object bounding-box diagonal leaves RADIANT at 27.1 and 28.5 dB, against RenderFormer’s 28.7 dB on clean input on the same images. Corrupting every entity (including high-coverage walls and floor) leads to steeper degradation because rasterized hit channels directly propagate geometric noise into the view decoder.

K ADDITIONAL RELATED WORK

Hierarchical and meshless transport. Hierarchical radiosity (Hanrahan et al., 1991) decomposes the form-factor matrix of n elements into $O(n)$ blocks by subdividing patches adaptively, and meshless hierarchical bases (Lehtinen et al., 2008) represent illumination independently of the geometric discretization for precomputed direct-to-indirect transport. Both compute transport for a given scene while RADIANT learns one mapping and applies it to unseen scenes.

Table 24: Zero-shot rendering from corrupted point inputs (no fine-tuning; RADIANT at step 600k; first 1,028 held-out images; PSNR per-image / LPIPS). *Objects*: only the foreground objects are corrupted; *All*: every entity, including walls, floor, and emitter geometry (emission values unchanged). On clean input the same images score 32.1 (RADIANT), 28.7 (RenderFormer), and 24.5 dB (RenderFormer-V2), all at step 600k.

Corruption	Level	Objects	All
Clean input	—	32.1 / 0.043	
Point density (samples per entity)	1,024	31.1 / 0.050	28.3 / 0.055
	512	27.1 / 0.079	25.2 / 0.082
	256	23.6 / 0.119	20.0 / 0.194
Position noise (σ , % of box diagonal)	0.5	30.8 / 0.049	23.5 / 0.078
	1	28.5 / 0.062	19.4 / 0.144
	2	25.4 / 0.095	15.8 / 0.266
Normal noise (mean angular error)	14°	31.4 / 0.054	25.0 / 0.133
	35°	29.7 / 0.076	19.7 / 0.309
	57°	—	15.7 / 0.376
Material noise (relative σ)	10%	—	30.6 / 0.056
	25%	—	27.5 / 0.090

Neural representations of global illumination. Neural radiance caching (Müller et al., 2021) adapts a network online inside a path tracer instead of pretraining it; active exploration (Diolatzis et al., 2022) trains a neural generator for one variable scene given an explicit parameterization of its configuration; neural vertex features (Su et al., 2025) store learnable features at mesh vertices, with an emphasis on neural radiosity; and object-centric feature grids model multi-frequency reflections in dynamic scenes (Mo et al., 2025). RADIANT is trained once across a million scenes and applied to unseen scenes without further training.

Point-based rendering and visibility. Surface splatting (Zwicker et al., 2001) renders point clouds without connectivity using a screen-space elliptical weighted average filter. The point splat of RADIANT is a simpler depth-tested splat with 7×7 dilation that supplies the primary G-buffer. Point-ersect (Chang et al., 2023) infers the intersection of a ray with the surface underlying a point cloud with a set transformer, which enables ray tracing of point clouds, including global illumination. In contrast, RADIANT traces no rays and predicts all lighting in one forward pass.

L LIMITATIONS, FUTURE WORK, AND BROADER IMPACT

Beyond the limitations of Sec. 5, the baselines are trained under our matched capacity and budget rather than their authors’ schedules; the fixed sample and token budget per object bounds the fine geometric detail that can be represented; and small scenes at batch size 1 favor triangle baselines before quadratic scaling dominates (Appendix E). This work uses only synthetic scenes assembled from publicly available Objaverse assets, involves no human subjects or personal data, and its intended use is faster physically based rendering; we do not foresee misuse specific to this method beyond that of rendering in general. The licenses of the assets, including the non-commercial terms of 1,093 of the 18,085 training meshes, are listed in the ethics statement.