LEDENOISER: EXPLORING LATENT WORLD MODELS FOR MONTE CARLO RENDERING DENOISING

Anonymous authors
Paper under double-blind review

ABSTRACT

Monte Carlo rendering converges as samples accumulate, yet deep learning-based denoising rarely models that process. The field has largely settled on image-to-image regression: spatial 2D filtering over rigid multi-channel inputs, supervised only on output radiance. We ask whether light transport convergence can instead be formulated as an evolving physical state transition. Inspired by latent world models, which learn action-conditioned dynamics entirely within a regularized representation space, we propose *LeDenoiser*, a latent world model for Monte Carlo denoising. A noisy 1-SPP observation forms the initial state, encoded by a pre-trained vision transformer; auxiliary Arbitrary Output Variables (AOVs) act as discrete token-level action operators; and the predictor learns an action-conditioned transition that advances this state toward convergence. On SPP-WORLD, our 22-scene multi-SPP path-tracing benchmark, unconditional latent dynamics lift severe 1-SPP noise (17.75 dB) to 32.61 dB, demonstrating that a latent world model can capture scene-level light transport without treating auxiliary geometry as a rigid dependency. Because AOVs form a discrete action space, that space can also be planned over. To our knowledge, this provides the first planning study inside a rendering world model: Model Predictive Control via the Cross-Entropy Method discovers sparse action subsets ($\bar{k} \approx 1.6$ of 8 buffers) that outperform exhaustive all-buffer conditioning. The latent dynamics are controllable, and geometric redundancy varies non-trivially across scenes. These findings establish latent world models as a viable alternative to image regression in Monte Carlo rendering, bridging graphics pipelines with physical foundation models.

1 INTRODUCTION

Monte Carlo (MC) rendering numerically evaluates the rendering equation by integrating stochastic light paths, producing photorealistic imagery across complex optical materials and illumination environments. However, generating cinematic-grade, hyper-realistic imagery requires massive sampling budgets. In complex scenes with intricate geometry or challenging indirect transport, convergence becomes considerably more difficult without sufficient path guiding, as raw estimators contend with high-variance paths that resist global noise reduction even under high sample counts. Because Monte Carlo variance decays slowly with sample count, relying purely on brute-force sampling to resolve residual noise is computationally prohibitive, establishing Monte Carlo denoising as an indispensable component in modern rendering pipelines.

Over the past decade, deep learning-based Monte Carlo denoising has seen rapid and impactful advancements. A diverse landscape of architectures has emerged: early learned methods predicted adaptive local filtering parameters or per-pixel reconstruction kernels, while subsequent works incorporated recurrent spatiotemporal aggregation, permutation-invariant path sample splatting, and deep multi-scale feature hierarchies that now power production engines. Across these technical variations, these approaches share a common operational foundation: they focus on learning robust reconstruction kernels, fusing spatiotemporal evidence, and leveraging AOVs—such as albedo, normals, and depth—within end-to-end, image-to-image regression frameworks. In other words, the field has largely converged on treating denoising as a spatial filtering problem over fixed input channels.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

World models, meanwhile, have emerged in Physical AI and reinforcement learning as a powerful paradigm for understanding dynamical systems. In particular, the LeWorldModel (LeWM) formulation demonstrates that an agent can learn action-conditioned dynamics entirely within a regularized latent representation space, supporting autonomous state-transition exploration and policy planning without the computational burden of pixel-level generation. Inspired by this formulation, we rethink the Monte Carlo denoising problem: can the convergence of light transport be conceptualized as an evolving physical state transition in a latent representation space?

We explore this idea through *LeDenoiser*, a latent world model for Monte Carlo denoising (Figure 1). The model operates within a vision transformer representation space: a noisy 1-SPP observation is encoded as the initial latent state, AOVs act as discrete action operators, and the predictor learns a single-step transition toward the converged reference state.

Contributions. Latent world model formulation. We formulate Monte Carlo denoising as an action-conditioned state transition within an image representation space. A world model built on these representations learns the convergence manifold of light transport, predicting how a noisy observation evolves toward a converged rendering. By exploiting the statistical equivalence inherent in progressive path tracing, the formulation collapses the multi-frame temporal sequence typical of world models into a single-step transition, substantially reducing memory and computation overhead.

AOVs as discrete/sparse actions. Within this framework, we treat AOVs as discrete action operators that condition the latent transition, replacing fixed-channel feature concatenation and spatial kernel filtering with a flexible action interface. Because each AOV is an independent discrete action, the model naturally supports sparse activation—accepting any subset of available buffers at inference time without retraining.

Latent planning in world model. We evaluate Model Predictive Control within the latent space of the world model. Using Cross-Entropy Method search over the AOV action space, the planner identifies sparse subsets (averaging 1.6 out of 8 buffers) that match or exceed exhaustive all-buffer conditioning, exposing scene-dependent geometric redundancy while indicating that multi-step rollout drift remains an open engineering bottleneck.

2 Related Work

2.1 Monte Carlo Denoising

Monte Carlo rendering estimates the rendering equation from stochastic paths; at low sample counts, denoising must suppress signal-dependent variance while preserving geometry, materials, and high-frequency illumination. Classical methods combine noisy radiance with albedo, normals, and depth, aggregating spatial or temporal evidence explicitly: à-trous filtering uses geometry-aware multiscale edge-stopping (Dammertz et al., 2010), RPF guides a cross-bilateral filter by random path parameters (Sen & Darabi, 2012), Bayesian collaborative denoising groups statistically similar neighborhoods (Boughida & Boubekeur, 2017), and SVGF combines temporal accumulation, variance estimation, and wavelet filtering (Schied et al., 2017). Learned methods replace fixed filtering rules with adaptive operators: predicting feature-filter parameters (Kalantari et al., 2015), per-pixel reconstruction kernels (Bako et al., 2017), recurrent aggregation of reprojected hidden states (Chaitanya et al., 2017), and permutation-invariant splatting of path samples (Gharbi et al., 2019). Their supervision nevertheless targets output radiance through direct image-space regression rather than modeling predictive scene states.

2.2 From JEPA to World Models

Self-supervised representation learning seeks transferable semantics in unlabeled observations while suppressing task-irrelevant detail. Unlike pixel-reconstruction objectives that model sensor noise and texture unconditionally, JEPA predicts the latent representation of a masked target from visible context while preventing representation collapse. I-JEPA established this paradigm for images (Assran et al., 2023), and the V-JEPA lineage extended it to motion-aware video features, scaled pretraining for action-conditioned robot planning, and dense self-supervision for spatial grounding

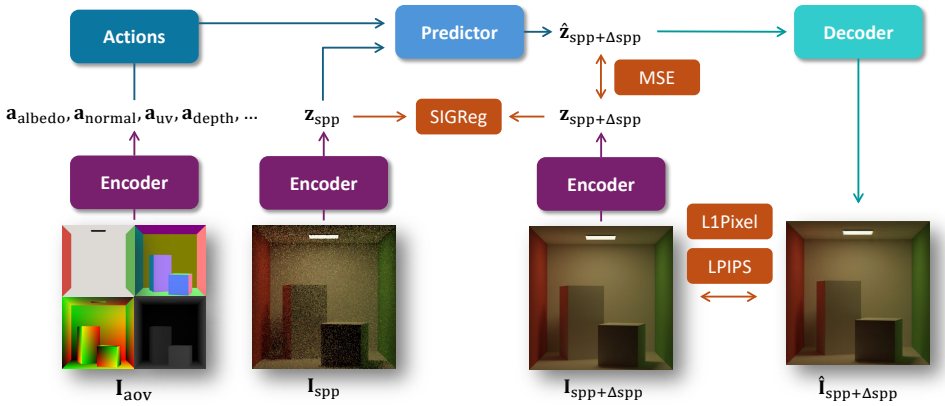

Figure 1: Overview of **LeDenoiser**. The three core components are a latent **predictor** for higher-SPP tokens, an **action** embedding module for AOV conditioning, and a pixel **decoder**. Given a low-SPP image $\mathbf{I}_t$ and a high-SPP target $\mathbf{I}_{t+\Delta t}$, the predictor infers the high-SPP latent $\hat{\mathbf{z}}_{t+\Delta t}$ conditioned on discrete action tokens $\mathbf{a}_t$ encoded from AOV buffers $\mathbf{I}_{\text{aov}}$. Predictor and decoder train either jointly or as independent modules.

(Bardes et al., 2025; Assran et al., 2025; Mur-Labadia et al., 2026). Beyond 2D data, Point-JEPA predicts ordered point-patch embeddings (Saito et al., 2025), 3D-JEPA combines multi-block sampling with a context-aware decoder (Hu et al., 2024), and D-JEPA couples autoregressive prediction with diffusion or flow-matching losses (Chen et al., 2024).

World models extend this by predicting action-conditioned dynamics for planning: whereas pixel generators model control-irrelevant detail, latent models predict compact states and evaluate candidate actions in feature space. LeWM brings this to end-to-end JEPA training (Maes et al., 2026), combining a compact visual encoder, an action-conditioned predictor, and SIGReg, which regularizes random embedding projections toward an isotropic Gaussian distribution (Maes et al., 2026; Bardes et al., 2022; Zbontar et al., 2021). A prediction loss and SIGReg suffice for learning from pixels and for cross-entropy-method planning in latent space. This motivates studying predictive latents as a complement to radiance reconstruction under Monte Carlo noise, extending world models from embodied physical environments to radiative transfer.

3 Method: LeDenoiser

We describe the three components of LeDenoiser—latent predictor, action embedding, and pixel decoder—and their training objectives; Figure 1 gives an overview.

3.1 Formulation and Training Objective

Latent prediction denoiser. Given the latent embedding of a noisy low-SPP observation $\mathbf{z}_t$, the predictor models Monte Carlo convergence in latent space by predicting the embedding of a higher-SPP rendered image $\hat{\mathbf{z}}_{t+\Delta t}$, conditioned on an auxiliary action sequence $\mathbf{a}_t$ corresponding to the same viewpoint (see Appendix B for a derivation from the rendering equation):

$$\begin{aligned} \text{Encoder}: &\quad \mathbf{z}_t = \text{Enc}(\mathbf{I}_t) \\ \text{Predictor}: &\quad \hat{\mathbf{z}}_{t+\Delta t} = \text{Pred}(\mathbf{z}_t, \mathbf{a}_t) \\ \text{Decoder}: &\quad \hat{\mathbf{I}}_{t+\Delta t} = \text{Dec}(\hat{\mathbf{z}}_{t+\Delta t}) \end{aligned} \tag{1}$$

Here $\text{Enc}(\cdot)$ is a pre-trained vision transformer (ViT) (Dosovitskiy et al., 2021; Oquab et al., 2024) that maps an image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ to patch tokens $\mathbf{z} \in \mathbb{R}^{N\times D}$, where N is the number of patches and D the embedding dimension, establishing a shared representation space between radiance and

AOVs. Keeping the encoder parameters fixed provides representational stability and prevents training collapse. We use layer-normalized patch tokens rather than deep global features, as denoising benefits from preserving both high-level semantics and spatial detail (see Appendix C and Table 3 for an empirical comparison across feature attributes). Only the predictor Pred, the action projection, and the decoder Dec are optimized during training. The predictor is shape-preserving ($\hat{\mathbf{z}}_{t+\Delta t} \in \mathbb{R}^{N\times D}$), and the decoder projects back to image space ($\hat{\mathbf{I}}_{t+\Delta t} \in \mathbb{R}^{3\times H\times W}$).

Predictor training. The predictor loss balances a prediction accuracy term and a sketched isotropic-Gaussian regularizer (SIGReg):

$$\mathcal{L}_{\mathrm{pred}} = \left\|\hat{\mathbf{z}}_{t+\Delta t} - \mathbf{z}_{t+\Delta t}\right\|_2^2 + \lambda_{\mathrm{reg}}\,\mathrm{SIGReg}\big(\hat{\mathbf{z}}_{t+\Delta t}\big), \tag{2}$$

where $\mathbf{z}_{t+\Delta t} = \mathrm{Enc}(\mathbf{I}_{t+\Delta t})$ is the target latent, and SIGReg is defined as:

$$\mathrm{SIGReg}\big(\hat{\mathbf{z}}_{t+\Delta t}\big) \triangleq \frac{1}{M}\sum_{m=1}^{M} T\big(\hat{\mathbf{z}}_{t+\Delta t}\,\mathbf{v}^{(m)}\big). \tag{3}$$

It projects the tokens onto M random unit directions $\mathbf{v}^{(m)}$ and averages a univariate normality statistic T over the resulting marginals. Following LeWM (Maes et al., 2026), SIGReg prevents the predicted tokens from collapsing onto a degenerate low-dimensional subspace.

Decoder training. The decoded prediction $\hat{\mathbf{I}}_{t+\Delta t}$ is supervised against the high-SPP reference by an ℓ_1 base fidelity term and an LPIPS perceptual term (Zhang et al., 2018):

$$\mathcal{L}_{\mathrm{dec}} = \left\|\hat{\mathbf{I}}_{t+\Delta t} - \mathbf{I}_{t+\Delta t}\right\|_1 + \lambda_{\mathrm{perc}}\,\mathcal{L}_{\mathrm{perc}}\big(\hat{\mathbf{I}}_{t+\Delta t},\,\mathbf{I}_{t+\Delta t}\big). \tag{4}$$

The ℓ_1 term is robust to outlier noise; $\mathcal{L}_{\mathrm{perc}}$ preserves perceptual texture.

Total objective. The combined objective optimizes predictor and decoder simultaneously:

$$\mathcal{L} = \mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{dec}}\mathcal{L}_{\mathrm{dec}}, \tag{5}$$

where λ_{dec} balances the two objectives. Here $\mathcal{L}_{\mathrm{pred}}$ anchors the state transition within the foundation latent manifold, while joint pixel supervision $\mathcal{L}_{\mathrm{dec}}$ provides complementary structural gradients to ensure that predicted states retain decodable high-frequency details. At test time, the state transition operates entirely in latent space, decoupling the dynamics model from pixel synthesis.

3.2 Action-Conditioned Latent Predictor

The predictor $\mathrm{Pred}_\theta(\mathbf{z}_t, \mathbf{a}_t)$ maps the noisy latent state $\mathbf{z}_t$ and action embeddings $\mathbf{a}_t$ to the refined target state. Before entering the predictor, $\mathbf{z}_t$ undergoes a linear projection and is augmented with learnable spatial positional embeddings, initialized as $\mathbf{z}^{(0)}$. The core architecture comprises L stacked dual-attention predictor blocks:

$$\begin{aligned}
\text{Self}: &\quad \mathbf{z}^{(l)\prime} = \mathbf{z}^{(l-1)} + \mathrm{MHSA}(\mathrm{LN}(\mathbf{z}^{(l-1)})) \\
\text{Cross}: &\quad \mathbf{z}^{(l)\prime\prime} = \mathbf{z}^{(l)\prime} + \mathrm{MHCA}\big(\mathrm{LN}(\mathbf{z}^{(l)\prime}), \mathrm{LN}(\mathbf{a}_t), \mathrm{LN}(\mathbf{a}_t)\big) \\
\text{FFN}: &\quad \mathbf{z}^{(l)} = \mathbf{z}^{(l)\prime\prime} + \mathrm{FFN}(\mathrm{LN}(\mathbf{z}^{(l)\prime\prime}))
\end{aligned} \tag{6}$$

Multi-Head Self-Attention (MHSA) models spatial dependencies across patch tokens. Multi-Head Cross-Attention (MHCA) injects action conditioning into the latent sequence. All sub-modules use pre-LN residual structures (Xiong et al., 2020). After L blocks, layer normalization and a linear projection yield the predicted latent state:

$$\hat{\mathbf{z}}_{t+\Delta t} = \mathrm{Proj}_{\mathrm{pred}}(\mathrm{LN}(\mathbf{z}^{(L)})). \tag{7}$$

3.3 AOVs as Discrete Action Operators

We formulate AOVs as discrete action operators that condition the latent state transition. Each AOV $\mathbf{I}_{\mathrm{aov}}$ is encoded by the same backbone into patch tokens, so that actions and image states share a common representation space.

Let $\mathcal{A} = \{(\mathbf{I}^{(i)}_{\text{aov}}, c_i)\}_{i=1}^{K}$ be the set of available AOVs, where c_i denotes the physical feature category (e.g., albedo, depth, geometric/shading normal, position, UV, primitive/shape index). Encoding yields $\mathbf{u}^{(i)} = \text{Enc}(\mathbf{I}^{(i)}_{\text{aov}}) \in \mathbb{R}^{N\times D}$. We augment this token grid with a learnable type embedding $\mathbf{e}_{c_i}$ and shared spatial positional embedding $\mathbf{P}$, projecting into action space:

$$\begin{aligned} \mathbf{a}^{(i)} &= \text{Proj}_{\text{act}}\Big(\mathbf{u}^{(i)} + \mathbf{e}_{c_i} + \mathbf{P}\Big), \\ \mathbf{a}_t &= \big[\,\mathbf{a}^{(1)}; \cdots ; \mathbf{a}^{(K)}\,\big] \in \mathbb{R}^{KN\times D}. \end{aligned} \tag{8}$$

The concatenated action sequence $\mathbf{a}_t$ serves as the key–value memory for cross-attention. Supplying an additional AOV simply appends N tokens to the action memory, so the predictor handles any AOV subset without retraining; the unconditional case ($K = 0$) uses a learnable null token $\mathbf{a}_\varnothing$. During training, K is randomly sampled from $\{0, \ldots, K_{\max}\}$, exposing the model to both conditional and unconditional modes. This discrete formulation naturally gives rise to a combinatorial action space over AOV subsets, which we exploit for latent planning in Section 4.4 (with extended analysis in Appendix D).

3.4 Pixel Decoder

The decoder Dec maps predicted latent tokens $\hat{\mathbf{z}}_{t+\Delta t} \in \mathbb{R}^{N\times D}$ back to pixel-space radiance $\hat{\mathbf{I}}_{t+\Delta t} \in \mathbb{R}^{3\times H\times W}$. Because the denoising task must produce an image, the decoder is necessary to verify that the latent state transition has arrived at a state corresponding to a clean rendering. To keep the readout direct without relying on heavy generative synthesis, we employ a lightweight convolutional-transformer structure.

Specifically, a sequence of Transformer blocks first processes the predicted tokens across S intermediate depths to produce multi-level features $\{\mathbf{h}^{(s)}\}_{s=1}^{S}$. A spatial reshaping operation ($\text{Grid} : \mathbb{R}^{N\times D} \to \mathbb{R}^{\sqrt{N}\times\sqrt{N}\times D}$) reassembles each token sequence onto the 2D patch grid. These representations are then progressively upsampled through residual convolutional blocks with lateral connections, followed by a final projection head:

$$\begin{aligned} \{\mathbf{h}^{(s)}\}_{s=1}^{S} &= \text{Refine}\big(\hat{\mathbf{z}}_{t+\Delta t}\big), \\ \hat{\mathbf{I}}_{t+\Delta t} &= \text{Head}\Big(\text{Up}\big(\{\text{Grid}(\mathbf{h}^{(s)})\}_{s=1}^{S}\big)\Big). \end{aligned} \tag{9}$$

Restricting the decoder to a direct structural readout ensures that evaluation metrics faithfully probe the state quality of the latent predictor rather than masking errors through decoder capacity. Architectural and hyperparameter details are provided in Appendix A.

4 Experiments

We evaluate the proposed framework along three experimental axes corresponding to our exploratory contributions: single-step state transition without geometric aids (C1), discrete action conditioning across arbitrary AOV subsets (C2), and latent-space planning via Model Predictive Control (C3).

4.1 Experimental Setup

The SPP-World benchmark. We assemble SPP-World, a path-traced dataset of 22 diverse scenes from Benedikt Bitterli's rendering resources (Bitterli, 2016). Using Mitsuba 3 (Jakob et al., 2022), we render thirteen progressive sample counts per scene ($2^0, 2^1, \ldots, 2^{12}$ SPP) at 336×336 resolution. Levels 1 through 2048 SPP form training pairs; independent 4096-SPP renders serve as held-out ground truth. The dataset includes 8 AOV modalities (17 channels): albedo, depth, geometric normal, position, shading normal, UV, primitive index, and shape index. Material BSDF parameters are excluded, requiring the model to infer light-surface interactions from radiance alone.

Per-scene dynamics modeling. Following the world-model paradigm in Physical AI—where an agent learns the dynamics of a specific environment—we fit one predictor–decoder pair per scene, allowing the model to learn that environment's light transport manifold. All experiments run on

Table 1: Quantitative evaluation on SPP-WORLD at 1-SPP input, measured against the held-out 4096-SPP reference. Baselines are OIDN (Áfra, 2021) (high-quality HDR mode) and OptiX (NVIDIA, 2026) (kernel-prediction mode), both provided with noise-free albedo and normal buffers on CUDA. We report our unconditional ($K = 0$) and fully conditioned ($K = 8$) variants; $K = 0$ achieves 32.61 dB without using any AOV. Best results in **bold**; second-best underlined.

	OIDN			OptiX			Ours (K=0)			Ours (K=8)		
Scene	PSNR ↑	SSIM ↑	FLIP ↓	PSNR ↑	SSIM ↑	FLIP ↓	PSNR ↑	SSIM ↑	FLIP ↓	PSNR ↑	SSIM ↑	FLIP ↓
coffee	21.77	0.4257	0.1618	22.95	0.5184	0.1386	33.51	0.9446	**0.0555**	**33.55**	**0.9449**	0.0557
cornell-box	27.61	0.5723	0.1261	28.05	0.5926	0.1158	38.50	0.9778	0.0424	**38.53**	**0.9779**	**0.0408**
dragon	27.88	0.9558	0.0244	28.98	**0.9587**	**0.0222**	32.44	0.8443	0.0376	**32.45**	0.8430	0.0374
glass-of-water	19.46	0.5486	0.1543	19.59	0.5557	0.1537	27.60	0.8792	0.0797	**27.73**	**0.8797**	**0.0779**
house	27.81	0.8731	0.0689	28.68	0.8855	0.0642	**32.89**	**0.9315**	**0.0624**	32.84	0.9311	0.0640
kitchen	15.36	0.1677	0.3211	17.92	0.3797	0.2438	30.55	0.8767	0.0902	**30.67**	**0.8775**	**0.0880**
lamp	21.00	0.2692	0.1941	21.30	0.3234	0.1676	**34.11**	**0.9651**	**0.0487**	34.08	0.9651	0.0491
lego	29.96	**0.9423**	0.0350	29.45	0.9377	**0.0349**	**33.88**	0.8265	0.0425	33.82	0.8252	0.0430
living-room	14.09	0.1582	0.4159	14.32	0.1806	0.3911	29.73	0.8798	0.1031	**29.81**	**0.8802**	**0.1018**
living-room-2	19.93	0.2624	0.2039	22.03	0.3997	0.1742	33.55	0.9170	0.0728	**33.62**	**0.9172**	**0.0719**
living-room-3	19.84	0.2293	0.2418	23.85	0.4567	0.1769	33.78	0.9136	0.0719	**33.82**	**0.9138**	**0.0711**
matpreview	20.15	0.2070	0.2119	21.44	0.2683	0.1965	38.89	0.9708	0.0525	**38.94**	**0.9710**	**0.0520**
rover	**31.41**	**0.9450**	0.0632	30.58	0.9352	**0.0596**	27.84	0.8883	0.0709	27.82	0.8876	0.0716
spaceship	24.22	0.5303	0.1144	24.09	0.4241	0.1154	**33.52**	**0.9576**	0.0411	33.51	0.9576	**0.0398**
staircase	23.68	0.5299	0.1680	25.25	0.6073	0.1497	31.13	0.8404	0.0852	**31.14**	**0.8406**	**0.0846**
staircase2	17.11	0.1873	0.3179	23.21	0.6234	0.1750	33.12	0.9013	0.0955	**33.21**	**0.9020**	**0.0912**
teapot	23.43	0.4441	0.1604	23.63	0.4555	0.1476	**39.14**	**0.9713**	0.0481	39.11	0.9712	**0.0479**
teapot-full	20.51	0.3488	0.2080	21.00	0.4015	0.1854	32.22	0.9323	0.0647	**32.25**	**0.9326**	**0.0645**
veach-ajar	8.64	0.0628	0.6561	8.93	0.0793	0.6056	24.23	0.6576	0.1899	**24.41**	**0.6595**	**0.1853**
veach-bidir	15.00	0.1225	0.3652	14.86	0.1185	0.3474	29.86	0.8177	0.1225	**30.10**	**0.8182**	**0.1167**
veach-mis	27.37	0.6667	0.0827	27.21	0.6497	0.0799	35.81	**0.9524**	0.0448	**35.81**	0.9522	**0.0442**
volumetric-caustic	17.62	0.1330	0.3636	17.45	0.1183	0.3392	31.21	0.8525	0.1205	**31.32**	**0.8528**	**0.1179**
Mean	21.54	0.4355	0.2118	22.49	0.4941	0.1856	32.61	0.8954	0.0746	**32.66**	**0.8955**	**0.0735**

a single NVIDIA RTX 4090. Complete path-tracing rendering protocols, architectural configurations, and training hyperparameters are provided in Appendix A. Evaluation metrics are PSNR (dB), SSIM (Wang et al., 2004), and FLIP (Andersson et al., 2020).

4.2 LATENT DENOISING PERFORMANCE

Training stability. Figure 2 confirms that training converges reliably across all 22 scenes: the latent prediction loss decreases monotonically while SIGReg maintains an isotropic equilibrium, and both pixel and perceptual decoder losses decline in lockstep. Furthermore, across all diverse benchmark scenes, the training of the latent world models exhibits a remarkably consistent and rapid convergence trajectory. Together, these dynamics demonstrate the stability and reliability of LeDenoiser's learning process.

Quantitative results. Table 1 reports 1-SPP denoising against held-out 4096-SPP references. From high-noise inputs averaging 17.75 dB, LeDenoiser achieves 32.61 dB unconditionally ($K = 0$) and 32.66 dB under full conditioning ($K = 8$). Production baselines OIDN (Áfra, 2021) and OptiX (NVIDIA, 2026), both provided with noise-free albedo and normal buffers, reach 21.54 dB and 22.49 dB respectively. The near-parity between $K = 0$ and $K = 8$ suggests that the world model effectively learns scene-level light transport and material rendering regularities: noisy patch tokens already exhibit an intrinsic capacity for macroscopic radiance reconstruction, successfully transitioning through the latent representation space toward low-noise target states.

Qualitative analysis. Figure 3 shows that the unconditional model recovers light propagation and scattering with notable accuracy: multi-bounce diffuse transfer, specular inter-reflections, and curved refractions are faithfully reproduced from a single 1-SPP observation. Nevertheless, two systematic visual artifacts are apparent. In *veach-mis* (Figure 3), the compact specular highlight of the point light source is absent from the reconstruction, and in *veach-bidir* (Figure 4), the glass sphere is lost entirely after decoding. Both failures involve small, high-contrast features that occupy only a few patch tokens; we attribute them primarily to the limited spatial detail capacity of the current lightweight decoder rather than to errors in latent prediction. Appendix E discusses decoder design refinements to address this.

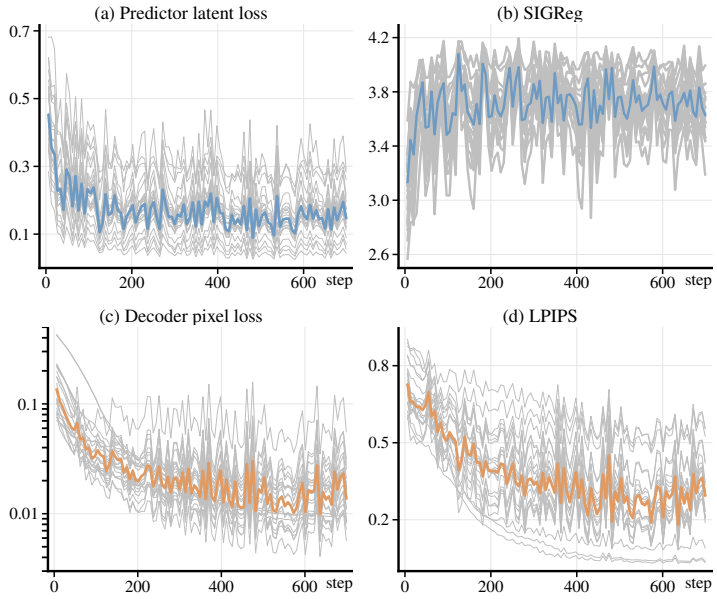

Figure 2: Training dynamics across all 22 scenes on SPP-WORLD (grey curves; median in colour). (a) Latent prediction loss converges steadily. (b) SIGReg stabilizes at an equilibrium plateau, confirming that manifold isotropy is maintained. (c, d) Decoder pixel loss and LPIPS decline monotonically, verifying that predicted latents decode into high-fidelity radiance.

Table 2: Latent-space planning evaluation on SPP-WORLD across 22 scenes (1-SPP input, horizon $H = 5$, execution block $B = 5$). CEM discovers that sparse AOV subsets outperform the exhaustive all-AOV baseline. Best results in **bold**.

Action Policy	PSNR ↑	SSIM ↑	FLIP ↓	Active AOVs ($\bar{k}$)
Fixed ($K = 0$, Unconditional)	30.13	0.8730	0.0917	0.0
Fixed ($K = 8$, Exhaustive)	30.16	0.8725	0.0922	8.0
CEM Active Search	**30.30**	**0.8745**	**0.0894**	**1.6**

4.3 CONDITIONING ON AOV SUBSETS

Unlike conventional pipelines where AOVs are concatenated into fixed input channels, our tokenized action formulation enables a single trained predictor to evaluate arbitrary subsets of AOVs without architectural changes or retraining.

Figure 4 shows qualitative reconstructions across conditioning cardinalities ($K \in \{0, 4, 8\}$). The model maintains consistent reconstruction across all levels. Unconditional prediction ($K = 0$) already recovers global radiance structures, while activating AOV actions ($K = 4, 8$) progressively refines fine-scale details such as caustic boundaries and specular highlights (e.g., in *veach-bidir*). Crucially, structuring conditioning along the token axis transforms AOV selection into an executable discrete action space, enabling the model to search for optimal feature combinations per scene.

4.4 LATENT-SPACE MPC PLANNING

Planning results and action sparsity. Treating AOVs as discrete action tokens allows test-time optimization via Model Predictive Control (Algorithm 1 in Appendix D). We run the Cross-Entropy Method (CEM) with a self-consistency objective that evaluates fixed-point residual stability without requiring reference ground truth. Table 2 reports aggregate metrics over the 22 benchmark scenes ($H = 5, B = 5$); per-scene numbers appear in Table 4. CEM finds sparse action schedules that outperform exhaustive conditioning (+0.14 dB over Fixed $K = 8$, improving on 16 of 22 scenes) while activating only 1.6 of 8 buffers on average ($\bar{k} \approx 1.6$). As detailed in Table 5, the optimizer isolates

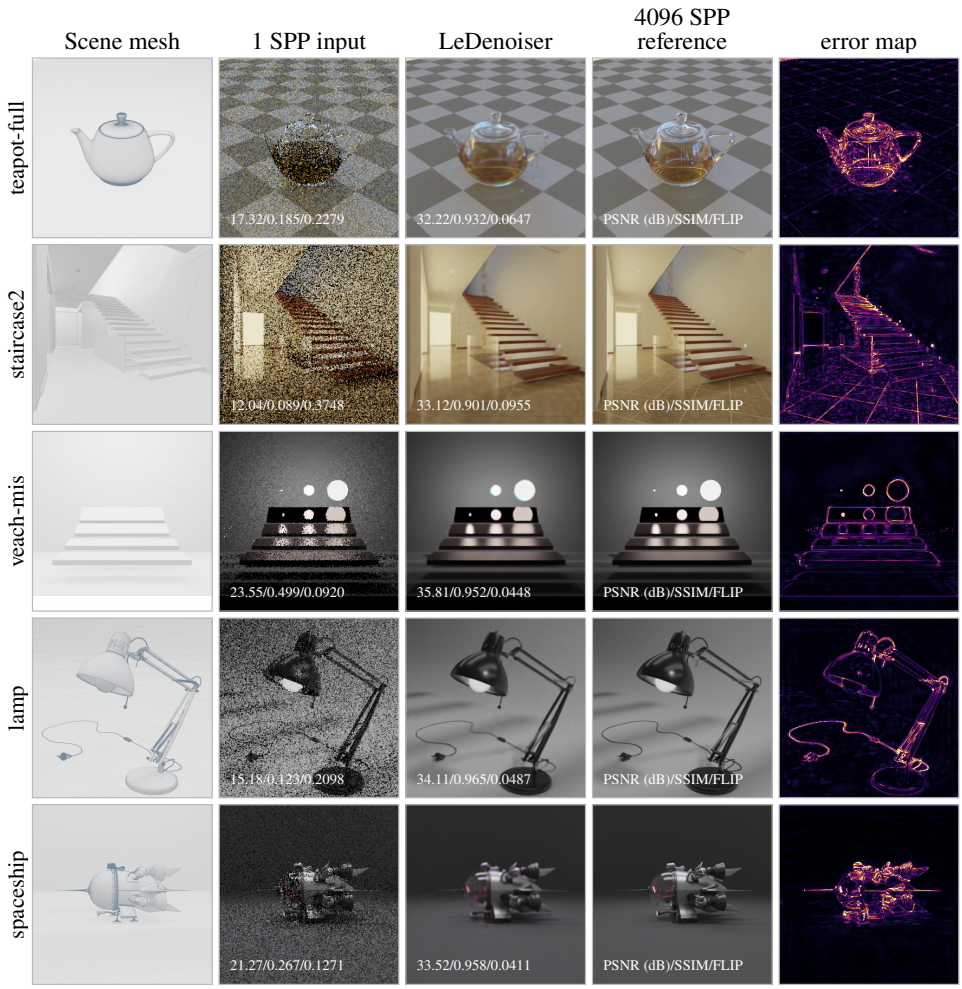

Figure 3: Qualitative results on five representative scenes at 1-SPP input. Left to right: the Scene mesh reference, 1-SPP input, LeDenoiser ($K = 0$), 4096-SPP ground truth, absolute error map. The model recovers smooth indirect illumination (*staircase2*), glossy reflections (*veach-mis*), and glass refraction (*teapot-full*) from a single noisy observation.

scene-specific geometric cues (e.g., planar normals on *cornell-box* vs. primitive IDs on *veach-mis*) rather than naively retaining all buffers, showing that the latent dynamics respond predictably to targeted action sequences.

Multi-step rollout drift. Although CEM finds sparse policies that surpass static conditioning, the 5-step rollout reaches 30.30 dB versus 32.66 dB for a single direct step ($K = 8$). This drop stems from compound autoregressive error accumulating over rollout horizons—a behavior well documented in physical world models. Addressing this drift through closed-loop replanning or policy distillation will be essential for integrating test-time search into practical interactive renderers.

5 Conclusion

We presented *LeDenoiser*, a latent world model-based Monte Carlo denoiser that recasts the denoising problem as an action-conditioned state transition in a learned representation space. By exploiting the statistical equivalence of progressive path tracing, the formulation collapses the multi-step temporal dynamics typical of world models into a single-step latent transition, keeping both memory and computation overhead low. Our experiments confirm three findings. First, training entirely

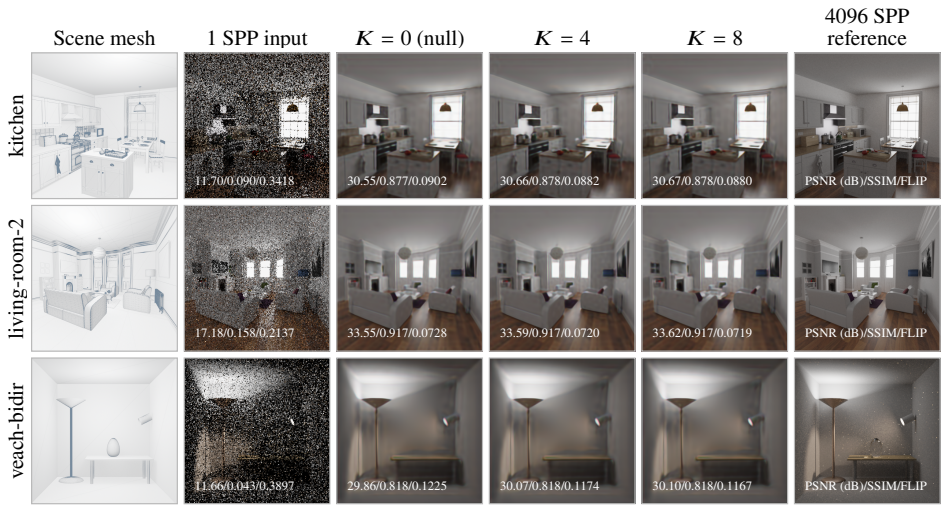

Figure 4: Visual effect of AOV conditioning cardinality on *kitchen*, *living-room-2*, and *veach-bidir* at 1 SPP. Left to right: Scene mesh, 1-SPP input, $K = 0$, $K = 4$, $K = 8$, and 4096-SPP reference, all generated using a single trained predictor. Insets show PSNR (dB) / SSIM / FLIP against the 4096-SPP reference. While unconditional prediction ($K = 0$) captures macroscopic illumination, discrete action conditioning ($K = 4, 8$) progressively refines high-frequency specular reflections and contact boundaries.

within vision-foundation representations, the model reconstructs high-quality images from extreme 1-SPP inputs and remains competitive with established pixel-domain denoisers. Second, treating AOVs as discrete action operators rather than fixed input channels allows any subset to be supplied at test time without retraining, providing per-scene flexibility absent in conventional architectures. Third, a Model Predictive Control search over this action space consistently identifies sparse AOV subsets—on average 1.6 out of 8 buffers—that match or exceed the quality obtained by exhaustive conditioning, exposing considerable geometric redundancy that varies across scenes. We discuss current limitations and directions for future work in Appendix E.

AI USE STATEMENT

In this work, generative AI tools were used for the following tasks: (1) assisting with academic translation and language editing to improve readability; (2) retrieving relevant literature and drafting an initial version of the Related Work section (Section 2); (3) code assistance for model implementation, experimental data aggregation, and plotting figures; and (4) checking intermediate algebraic steps in the temporal dimension collapse derivation in Appendix B, while all theoretical concepts, claims, and derivations were originally formulated by the authors.

Generative AI tools were not used for generating synthetic datasets, developing theoretical frameworks, proposing hypotheses, designing experiments, or interpreting results. Tasks related to human subjects or qualitative analysis are not applicable.

All AI-assisted content, derivations, and code were reviewed and verified by the authors. Citations were checked against original publications, and all experimental results were obtained by running the verified code. The authors take full responsibility for the contents of this paper.

ETHICS STATEMENT

This work investigates denoising of synthetically rendered computer graphics. It involves no human subjects, no personally identifiable data, and no private datasets: SPP-WORLD is derived solely from publicly released scenes (Bitterli, 2016) in accordance with their respective licenses, rendered via an open-source path tracer. We foresee no direct pathway from this foundational methodology to

surveillance, disinformation, or other harmful applications beyond the general dual-use character of computer vision research. The authors declare no competing financial interests.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, full architectural specifications, optimization hyperparameters, loss configurations, and complete path-tracing rendering protocols are detailed in Appendix A. The SPP-WORLD benchmark dataset and progressive sampling protocol are outlined in Section 4.1. Formal mathematical derivations for our state transition formulation are provided in Appendix B, and complete algorithmic details for Model Predictive Control planning appear in Appendix D.

REFERENCES

Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D. Fairchild. FLIP: A difference evaluator for alternating images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 3(2):15:1–15:23, 2020. doi: 10.1145/3406183. Proc. High Performance Graphics.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023. doi: 10.1109/CVPR52729.2023.01501.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. doi: 10.48550/arXiv.2506.09985.

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

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

Adrien Bardes, Quentin Garrido, Jean Ponce, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. In *International Conference on Learning Representations*, 2025.

Benedikt Bitterli. Rendering resources, 2016. https://benedikt-bitterli.me/resources/.

Malik Boughida and Tamy Boubekeur. Bayesian collaborative denoising for monte carlo rendering. *Computer Graphics Forum*, 36(4):137–153, 2017. doi: 10.1111/cgf.13231.

Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. Interactive reconstruction of monte carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics*, 36(4):98:1–98:12, 2017. doi: 10.1145/3072959.3073601.

Dengsheng Chen, Jie Hu, Xiaoming Wei, and Enhua Wu. Denoising with a joint-embedding predictive architecture. *arXiv preprint arXiv:2410.03755*, 2024. doi: 10.48550/arXiv.2410.03755.

Holger Dammertz, Daniel Sewtz, Johannes Hanika, and Hendrik P. A. Lensch. Edge-avoiding à-trous wavelet transform for fast global illumination filtering. In *Proceedings of High Performance Graphics*, pp. 67–75, 2010. doi: 10.2312/EGGH/HPG10/067-075.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Michaël Gharbi, Tzu-Mao Li, Miika Aittala, Jaakko Lehtinen, and Frédo Durand. Sample-based monte carlo denoising using a kernel-splatting network. *ACM Transactions on Graphics*, 38(4): 125:1–125:12, 2019. doi: 10.1145/3306346.3322954.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15979–15988, 2022. doi: 10.1109/CVPR52688.2022.01553.

Naiwen Hu, Haozhe Cheng, Yifan Xie, Shiqi Li, and Jihua Zhu. 3D-JEPA: A joint embedding predictive architecture for 3d self-supervised representation learning. *arXiv preprint arXiv:2409.15803*, 2024. doi: 10.48550/arXiv.2409.15803.

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. Mitsuba 3 renderer, 2022. https://mitsuba-renderer.org.

Nima Khademi Kalantari, Steve Bako, and Pradeep Sen. A machine learning approach for filtering monte carlo noise. *ACM Transactions on Graphics*, 34(4):122:1–122:12, 2015. doi: 10.1145/2766977.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026. doi: 10.48550/arXiv.2603.19312.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Michael Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026. doi: 10.48550/arXiv.2603.14482.

NVIDIA. NVIDIA OptiX® Ray Tracing Engine. `https://developer.nvidia.com/rtx/ray-tracing/optix`, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=a68SUt6zFt`.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12159–12168, 2021. doi: 10.1109/ICCV48922.2021.01196.

Ayumu Saito, Prachi Kudeshia, and Jiju Poovvancheri. Point-JEPA: A joint embedding predictive architecture for self-supervised learning on point cloud. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 7348–7357, 2025.

Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty Reddy Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron Lefohn, and Marco Salvi. Spatiotemporal variance-guided filtering: Real-time reconstruction for path-traced global illumination. In *Proceedings of High Performance Graphics*, pp. 1–12, 2017. doi: 10.1145/3105762.3105770.

Pradeep Sen and Soheil Darabi. On filtering the noise from the random parameters in monte carlo rendering. *ACM Transactions on Graphics*, 31(3):18:1–18:15, 2012. doi: 10.1145/2167076.2167083.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International conference on machine learning*, pp. 10524–10533. PMLR, 2020.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12310–12320, 2021.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, 2017. doi: 10.1109/TCI.2016.2644865. Published online 2016.

Attila T. Áfra. Intel® Open Image Denoise, 2021.

A Additional Implementation Details

This appendix provides extended derivations, architectural analyses, and experimental data supporting the main manuscript: a formal proof of temporal dimension collapse (Appendix B), an empirical analysis of latent representation attributes (Appendix C), the comprehensive latent-space planning study (Appendix D), and an expanded discussion of limitations and future work (Appendix E).

Benchmark and Path-Tracing Configuration. The SPP-WORLD benchmark comprises 22 diverse scenes from Benedikt Bitterli's rendering resources (Bitterli, 2016), rendered using Mitsuba 3 (Jakob et al., 2022) (variant `cuda_ad_rgb` with CPU fallback to `llvm_ad_rgb`) with the standard path tracer (integrator `path`, maximum path depth 12, Russian roulette enabled). All scenes are rendered at 336×336 square resolution in linear RGB (32-bit floating-point OpenEXR, without gamma compression or tonemapping; preview RGB images are generated via Reinhard tonemapping with gamma 2.2). For each scene, an independent sampler generates twelve progressive sampling levels ($2^0, 2^1, \ldots, 2^{11}$ SPP) sharing a continuous pseudo-random stream, guaranteeing the progressive sample accumulation property formalized in Appendix B. Independent 4096-SPP renders (2^{12} SPP) serve as held-out ground truth. The 8 AOV modalities (17 channels: albedo, depth, geometric normal, position, shading normal, UV, primitive index, and shape index) are extracted via Mitsuba's AOV integrator wrapper. Training pairs are formed by drawing input levels from $\{2^0, 2^1, \ldots, 2^{10}\}$ SPP paired with the 2048-SPP target. At each training step, conditioning cardinality k is sampled cyclically from $\{0, \ldots, K_{\max}\}$ ($K_{\max} = 8$); the AOV modalities are randomly shuffled and the first k modalities are selected, ensuring that the unconditional mode $k = 0$ occupies a balanced $1/(K_{\max} + 1) = 1/9$ training probability.

Model Architecture Specifications. We use a pre-trained ViT-B/14 backbone (Oquab et al., 2024) processing 336×336 inputs into a 24×24 grid of $N = 576$ patch tokens of dimension $D = 768$. The latent predictor comprises $L = 6$ Transformer blocks with 12 attention heads ($d_{\text{head}} = 64$), FFN hidden dimension 2048, dropout 0.1, and Pre-LN residual connections. The pixel decoder (111.9M parameters) consists of a hierarchical token refinement module with 8 Transformer layers and 32 attention heads ($d_{\text{head}} = 24$), extracting multi-depth feature taps at depths $\{2, 4, 6, 8\}$ ($S = 4$). These features are mapped by a spatial upsampling decoder featuring 4 ResNet consistency blocks (channel dim 768), 4 hierarchical skip projections (dim 768 to 768, 512, 256, 128), and 5 progressive convolutional upsampling stages with bilinear interpolation (channels $768 \to 512 \to 256 \to 128 \to 64 \to 32$) upscaling from 24×24 to 336×336, followed by an RGB projection head with sigmoid activation.

Training and Loss Hyperparameters. Training uses the AdamW optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$, weight decay 10^{-3}) with a constant learning rate of 5×10^{-5} for 100 epochs (700 optimization steps per scene) with batch size 4 in 16-bit mixed precision on a single NVIDIA RTX 4090 GPU (24 GB VRAM). In the total objective (Eq. 5), the pixel-loss balancing factor is set to $\lambda_{\text{dec}} = 1.0$, while the component loss weights in Eq. 2 and Eq. 4 are $\lambda_{\text{reg}} = 0.09$ and $\lambda_{\text{perc}} = 0.1$. The SIGReg regularizer (Eq. 3) projects predicted tokens onto $M = 1024$ random unit vectors $\mathbf{v}^{(m)} \sim \mathcal{S}^{D-1}$, where statistic T evaluates the Epps-Pulley empirical characteristic function discrepancy across 17 knots ($t \in [0, 3]$) against a standard normal distribution $\mathcal{N}(0, 1)$.

Planning Hyperparameters. The planning experiments in Section 4.4 and Appendix D use execution horizon T=5, planning horizon H=5, execution block B=5 (open-loop planning), momentum α=0.1, and an action space of K=8 AOV gate logits. The cross-entropy solver utilizes a population of C=300 candidate action sequences, I=30 refinement iterations, and an elite set of M=30. Predictor and decoder parameters remain fixed during planning; no gradient backpropagation occurs during action search.

B Temporal Dimension Collapse

In the main text, we formulate denoising as an action-conditioned state transition $\hat{\mathbf{z}}_{t+\Delta t} = \text{Pred}(\mathbf{z}_t, \mathbf{a}_t)$, where the predictor consumes only the current noisy state $\mathbf{z}_t = \text{Enc}(\mathbf{I}_t)$ and action tokens $\mathbf{a}_t$, without conditioning on historical latent states $(\mathbf{z}_{t-1}, \mathbf{z}_{t-2})$. We refer to this property as

temporal collapse. Here, we provide a formal derivation demonstrating that this collapse is an exact mathematical consequence of Monte Carlo integration.

B.1 Monte Carlo Rendering Formulation

The rendering equation describes outgoing radiance L_o at surface point $\mathbf{p}$ along direction ω_o:

$$L_o(\mathbf{p}, \omega_o) = L_e(\mathbf{p}, \omega_o) + \int_\Omega f_r(\mathbf{p}, \omega_i, \omega_o)\, L_{\text{in}}(\mathbf{p}, \omega_i)\, (\mathbf{n} \cdot \omega_i)\, d\omega_i \tag{10}$$

where L_e is emission, f_r is the BRDF, L_{in} is incident radiance, $\mathbf{n}$ is the surface normal, and integration covers hemisphere Ω. Because L_{in} is itself governed by Eq. 10 at the preceding intersection, the integral is recursive and admits no closed form.

Monte Carlo integration approximates it by drawing N stochastic samples $\{\omega_i\}_{i=1}^N$ from a density $p(\omega)$:

$$L_o(\mathbf{p}, \omega_o) \approx L_e(\mathbf{p}, \omega_o) + \frac{1}{N} \sum_{i=1}^{N} \frac{f_r(\mathbf{p}, \omega_i, \omega_o)\, L_{\text{in}}(\mathbf{p}, \omega_i)\, (\mathbf{n} \cdot \omega_i)}{p(\omega_i)}. \tag{11}$$

Absorbing the emission term, we define the radiance carried by the i-th path as

$$C_i \triangleq L_e(\mathbf{p}, \omega_o) + \frac{f_r(\mathbf{p}, \omega_i, \omega_o)\, L_{\text{in}}(\mathbf{p}, \omega_i)\, (\mathbf{n} \cdot \omega_i)}{p(\omega_i)}, \tag{12}$$

where L_{in} is estimated recursively by the same construction along the continuation of the path. Since $\frac{1}{N}\sum_{i=1}^N L_e = L_e$, Eq. 11 is then exactly the sample mean

$$L_N = \frac{1}{N} \sum_{i=1}^{N} C_i. \tag{13}$$

Taking $\mathbf{p}$ and ω_o to be the primary intersection and reverse view direction of a camera ray, L_N is the estimate at the corresponding pixel. Here N denotes Samples Per Pixel (SPP), and $\mathbf{I}_t$ collects L_N over all pixels at sampling stage N.

B.2 Batch–Progressive Equivalence

We prove that batch rendering (evaluating all N samples simultaneously) and progressive rendering (accumulating samples sequentially via online averaging) are mathematically equivalent under a shared pseudorandom sequence.

Proof by Induction. Let $\{r_1, r_2, \dots\}$ be the shared PRNG sequence generating path directions $\{\omega_1, \omega_2, \dots\}$. Define the batch estimator as $L_{\text{batch}}(n) = \frac{1}{n}\sum_{i=1}^n C_i$ and the progressive online estimator as $L_{\text{prog}}(n)$.

Base case ($n = 1$): $L_{\text{prog}}(1) = C_1 = L_{\text{batch}}(1)$.

Inductive step ($m \to m+1$): Assume $L_{\text{prog}}(m) = L_{\text{batch}}(m) = \frac{1}{m}\sum_{i=1}^m C_i$. The progressive update incorporates sample C_{m+1}:

$$\begin{aligned} L_{\text{prog}}(m+1) &= L_{\text{prog}}(m) + \frac{C_{m+1} - L_{\text{prog}}(m)}{m+1} \\ &= \frac{m\, L_{\text{prog}}(m) + C_{m+1}}{m+1} = \frac{\sum_{i=1}^m C_i + C_{m+1}}{m+1} = L_{\text{batch}}(m+1). \end{aligned} \tag{14}$$

By mathematical induction, $L_{\text{prog}}(n) = L_{\text{batch}}(n)$ for all $n \geq 1$. □

B.3 Derivation of History-Independent State Transitions

Progressing from N SPP ($\mathbf{I}_t$) to $N + M$ SPP ($\mathbf{I}_{t+\Delta t}$) follows the exact recurrence:

$$L_{N+M} = \frac{N}{N+M}\, L_N + \frac{M}{N+M}\, L_{M_{\text{new}}} \tag{15}$$

Table 3: Cosine similarity comparison of DINOv2 feature attributes across SPP levels against 2048-SPP references (ViT-B/14 with register tokens). Patch tokens preserve the dense spatial grid while maintaining monotonic sensitivity to Monte Carlo noise levels.

Scene	Feature Attribute	1 spp	4 spp	16 spp	64 spp	256 spp	1024 spp
coffee	**Patch Tokens (2D)**	**0.666**	**0.763**	**0.827**	**0.895**	**0.937**	**0.956**
	Register Tokens (1D)	0.922	0.945	0.959	0.981	0.992	0.999
	Class Token (1D)	0.834	0.916	0.953	0.979	0.993	0.998
	Mean-Pooled (1D)	0.792	0.870	0.913	0.958	0.985	0.996
	PreNorm Tokens	0.738	0.806	0.857	0.911	0.947	0.963
cornell-box	**Patch Tokens (2D)**	**0.709**	**0.772**	**0.842**	**0.894**	**0.932**	**0.959**
	Register Tokens (1D)	0.844	0.880	0.942	0.978	0.993	0.998
	Class Token (1D)	0.760	0.856	0.925	0.965	0.988	0.997
	Mean-Pooled (1D)	0.824	0.872	0.930	0.967	0.989	0.998
	PreNorm Tokens	0.786	0.829	0.882	0.922	0.950	0.969
dragon	**Patch Tokens (2D)**	**0.684**	**0.861**	**0.937**	**0.981**	**0.987**	**0.998**
	Register Tokens (1D)	0.779	0.893	0.960	0.988	0.993	0.999
	Class Token (1D)	0.336	0.772	0.913	0.963	0.987	0.997
	Mean-Pooled (1D)	0.692	0.884	0.947	0.989	0.988	0.999
	PreNorm Tokens	0.707	0.873	0.942	0.984	0.988	0.999
veach-mis	**Patch Tokens (2D)**	**0.729**	**0.795**	**0.859**	**0.905**	**0.928**	**0.958**
	Register Tokens (1D)	0.933	0.948	0.975	0.986	0.994	0.998
	Class Token (1D)	0.764	0.854	0.936	0.963	0.984	0.994
	Mean-Pooled (1D)	0.856	0.899	0.954	0.978	0.990	0.997
	PreNorm Tokens	0.796	0.845	0.897	0.932	0.949	0.971
teapot-full	**Patch Tokens (2D)**	**0.601**	**0.800**	**0.863**	**0.908**	**0.938**	**0.961**
	Register Tokens (1D)	0.826	0.947	0.968	0.982	0.991	0.997
	Class Token (1D)	0.474	0.888	0.946	0.966	0.986	0.996
	Mean-Pooled (1D)	0.713	0.899	0.938	0.966	0.985	0.996
	PreNorm Tokens	0.683	0.844	0.895	0.931	0.955	0.972

where $L_{M_{\text{new}}}$ is the average of M newly sampled paths. Eq. 15 holds independently at every pixel, so it applies elementwise to the full image, with $\mathbf{I}_{t+\Delta t}$ the same convex combination of $\mathbf{I}_t$ and the image formed by the M new samples. Because Monte Carlo paths are independent and identically distributed, their evaluation order is arbitrary, collapsing the physical temporal dimension. Consequently, observation $\mathbf{I}_t$ forms a sufficient statistic for all prior samples. Predicting $\mathbf{I}_{t+\Delta t}$ reduces to inferring the expectation of the incremental radiance integral, justifying a direct, history-independent state transition:

$$\hat{\mathbf{z}}_{t+\Delta t} = \text{Pred}(\mathbf{z}_t, \mathbf{a}_t). \tag{16}$$

This derivation formally explains why LeDenoiser can discard historical observation buffers $(\mathbf{z}_{t-1}, \mathbf{z}_{t-2})$ that are typically required by recurrent video world models.

C Choice of Latent Representation

Choosing an appropriate representation from the DINOv2 backbone is crucial. A forward pass yields class tokens, register tokens, patch tokens, and mean-pooled global features. A valid denoising state representation must satisfy two criteria: (1) preserving 2D spatial structure, and (2) exhibiting monotonic sensitivity to Monte Carlo variance.

Table 3 reports cosine similarities between noisy inputs (1 to 1024 SPP) and 2048-SPP references across five representative scenes. Register tokens exhibit artificial noise invariance (0.78 to 0.93 similarity even at 1 SPP), masking high-frequency variance and preventing the dynamics model from learning convergence trajectories. Global embeddings (class token and mean-pooled patches) collapse 2D spatial structure. In contrast, patch tokens preserve the 2D spatial grid while demonstrating monotonic, noise-sensitive similarity scaling (scaling steadily from 0.60–0.73 at 1 SPP to ~ 0.96 at 1024 SPP across scenes, reaching 0.998 on *dragon*), establishing them as the ideal representation for spatial state prediction.

Algorithm 1 Model Predictive Control with Latent Planning

Require: Initial noisy tokens $\mathbf{z}_0 \in \mathbb{R}^{N\times D}$, trained predictor Pred, cost function $\mathcal{J}$
Require: Horizon H, execution block B, total steps T, population C, elite size M, iterations I, momentum α
Ensure: Denoised tokens $\hat{\mathbf{z}}_{\text{final}}$, executed plan $(\mathbf{a}_1^*, \ldots, \mathbf{a}_T^*)$

$\mathbf{z} \leftarrow \mathbf{z}_0$, $t \leftarrow 0$, $\boldsymbol{\mu} \leftarrow \mathbf{0}_{H\times K}$, $\boldsymbol{\sigma} \leftarrow \mathbf{1}_{H\times K}$
while $t < T$ **do**
  // — Phase 1: CEM optimizes H-step actions —
  for $i = 1$ **to** I **do**
    Sample $\{\boldsymbol{\epsilon}^{(c)}\}_{c=1}^{C} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_{H\times K})$
    $\mathbf{a}^{(c)} \leftarrow \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}^{(c)}$, $c = 1, \ldots, C$
    for $c = 1$ **to** C **do**
      $\hat{\mathbf{z}} \leftarrow \mathbf{z}$ *// World Model: rollout*
      for $\tau = 1$ **to** H **do**
        $\hat{\mathbf{z}} \leftarrow \mathrm{Pred}\big(\hat{\mathbf{z}},\, \mathrm{Gate}_{\mathrm{hard}}(\mathbf{a}_\tau^{(c)})\big)$
      end for
      $J^{(c)} \leftarrow \mathcal{J}\big(\hat{\mathbf{z}},\, \mathbf{a}_H^{(c)}\big)$ *// Cost: score terminal state*
    end for
    $\pi \leftarrow \mathrm{argsort}(J^{(1)}, \ldots, J^{(C)})$ *// Elite selection*
    $\hat{\boldsymbol{\mu}} \leftarrow \frac{1}{M}\sum_{j=1}^{M} \mathbf{a}^{(\pi(j))}$, $\hat{\boldsymbol{\sigma}} \leftarrow \mathrm{std}\big(\{\mathbf{a}^{(\pi(j))}\}_{j=1}^{M}\big)$
    $\boldsymbol{\mu} \leftarrow (1-\alpha)\,\hat{\boldsymbol{\mu}} + \alpha\,\boldsymbol{\mu}$, $\boldsymbol{\sigma} \leftarrow (1-\alpha)\,\hat{\boldsymbol{\sigma}} + \alpha\,\boldsymbol{\sigma}$
  end for
  $\mathbf{a}^* \leftarrow \boldsymbol{\mu}$
  // — Phase 2: Execute first B steps —
  $B_{\text{exec}} \leftarrow \min(B,\, T - t)$
  for $\tau = 1$ **to** B_{exec} **do**
    $\mathbf{z} \leftarrow \mathrm{Pred}\big(\mathbf{z},\, \mathrm{Gate}_{\mathrm{hard}}(\mathbf{a}_\tau^*)\big)$
  end for
  // — Phase 3: Warm-start next window —
  $\boldsymbol{\mu} \leftarrow (\mathbf{a}^*_{B_{\text{exec}}+1}, \ldots, \mathbf{a}^*_H, \underbrace{\mathbf{a}^*_H, \ldots, \mathbf{a}^*_H}_{B_{\text{exec}}})$
  $\boldsymbol{\sigma} \leftarrow \mathbf{1}_{H\times K}$
  $t \leftarrow t + B_{\text{exec}}$
end while
return $\mathbf{z}$, executed plan

D Extended Latent Planning Study

This section details the Model Predictive Control (MPC) planning framework, provides algorithmic descriptions, and reports per-scene breakdowns of discovered action plans. Visual evidence across conditioning cardinalities is presented in Figure 4 of the main text.

Algorithm 1 outlines the MPC framework for latent rendering dynamics. The predictor advances noisy patch tokens $\mathbf{z}_t$ under action gates $\mathbf{a}_t \in \mathbb{R}^K$, mapped to binary execution masks via $\mathrm{Gate}_{\mathrm{hard}}(\mathbf{a}) = \mathbb{I}(\mathbf{a} > 0) \in \{0,1\}^K$ to select the active AOV modalities. Because reference images are unavailable at inference time, candidate sequences are scored by a self-consistency cost $\mathcal{J}\big(\hat{\mathbf{z}}_H, \mathbf{a}_H\big) = \sum_{n,d}\big(\mathrm{Pred}(\hat{\mathbf{z}}_H, \mathbf{a}_H)_{n,d} - \hat{\mathbf{z}}_{H,n,d}\big)^2$, which measures the fixed-point residual of the terminal latent state.

D.1 Per-Scene Breakdown

Table 4 gives the scene-by-scene breakdown. CEM surpasses Fixed ($K = 8$) on 16 of 22 scenes, with the largest margins occurring in specular and glossy configurations: *veach-mis* (+0.78 dB), *matpreview* (+0.44 dB), *teapot-full* (+0.41 dB), and *cornell-box* (+0.40 dB).

Table 4: Detailed per-scene planning results on SPP-WORLD (1 SPP $\rightarrow$ 4096 SPP, $H = 5$). CEM achieves superior fidelity on 16/22 scenes while maintaining high action sparsity ($\bar{k} \approx 1.6$).

	CEM				Random				Fixed (K=8)			Fixed (K=0)		
Scene	PSNR↑	SSIM↑	FLIP↓	$\bar{k}$	PSNR↑	SSIM↑	FLIP↓	$\bar{k}$	PSNR↑	SSIM↑	FLIP↓	PSNR↑	SSIM↑	FLIP↓
coffee	**31.97**	**0.936**	**0.076**	1.2	31.78	0.934	0.079	3.6	31.78	0.934	0.080	31.74	0.934	0.077
cornell-box	**37.29**	0.974	**0.055**	2.4	37.21	**0.975**	0.056	3.6	36.89	0.974	0.060	36.83	0.974	0.063
dragon	29.39	0.826	**0.047**	3.2	29.51	**0.827**	**0.047**	4.6	**29.67**	0.825	0.048	29.41	0.822	0.049
glass-water	**27.42**	**0.872**	**0.086**	1.4	27.25	0.869	0.090	2.6	27.12	0.867	0.092	27.32	0.870	0.088
house	30.20	**0.909**	**0.088**	2.6	30.13	0.908	0.089	3.6	29.90	0.905	0.093	**30.26**	**0.909**	0.089
kitchen	**27.06**	**0.829**	**0.116**	1.8	27.05	0.828	0.118	3.0	27.03	0.828	0.119	26.85	0.826	0.118
lamp	29.43	**0.934**	**0.074**	0.8	29.34	0.933	0.075	2.4	**29.51**	**0.934**	0.075	29.29	0.933	0.075
lego	**31.64**	**0.812**	**0.054**	1.4	31.62	0.811	0.055	3.6	31.56	0.809	0.056	31.55	0.811	0.056
living-room	26.51	**0.836**	**0.136**	2.2	26.49	**0.836**	0.138	2.4	**26.54**	**0.836**	**0.136**	26.36	0.834	0.140
living-room-2	**31.09**	**0.896**	**0.092**	1.8	31.08	0.895	**0.092**	3.6	30.91	0.894	0.094	30.91	0.894	0.095
living-room-3	31.04	0.889	**0.086**	1.4	31.11	0.889	0.087	2.2	**31.14**	**0.890**	0.087	30.99	0.888	0.088
matpreview	35.09	**0.956**	**0.071**	1.6	34.85	0.955	0.073	3.2	34.65	0.953	0.075	**35.11**	**0.956**	0.072
rover	**25.69**	**0.855**	**0.097**	2.2	25.65	0.853	0.099	3.6	25.52	0.850	0.101	25.49	0.854	0.098
spaceship	**31.18**	**0.944**	**0.056**	0.4	31.10	0.943	0.061	3.6	31.09	0.943	0.059	30.58	0.940	0.058
staircase	**29.87**	**0.821**	**0.104**	1.8	**29.87**	0.820	0.105	2.4	29.76	0.819	0.107	29.77	0.819	0.105
staircase2	**29.98**	**0.877**	**0.110**	0.6	29.93	0.876	0.111	2.4	29.80	0.874	0.113	29.80	0.875	0.113
teapot	33.94	0.940	**0.071**	1.4	**34.05**	**0.941**	**0.071**	2.4	33.91	0.940	0.073	33.93	0.940	0.072
teapot-full	**30.60**	**0.908**	**0.081**	1.4	30.32	0.904	0.085	2.4	30.19	0.902	0.087	30.20	0.905	0.084
veach-ajar	**24.50**	**0.617**	**0.178**	2.0	24.36	0.613	0.183	2.4	24.26	0.609	0.185	24.45	0.614	0.180
veach-bidir	28.98	0.809	0.112	1.8	29.05	**0.810**	**0.110**	4.0	**29.06**	**0.810**	**0.110**	28.79	0.808	0.116
veach-mis	**32.95**	**0.947**	**0.062**	0.6	32.20	0.945	0.071	2.6	32.17	0.944	0.072	32.71	0.946	0.064
vol.-caustic	30.72	0.852	0.113	0.6	30.78	0.852	0.111	3.2	**31.06**	**0.853**	**0.109**	30.60	0.852	0.117
Average	**30.30**	**0.875**	**0.089**	**1.6**	30.21	0.874	0.091	3.1	30.16	0.873	0.092	30.13	0.873	0.092

D.2 DISCOVERED ACTION PLANS

Table 5 reports the per-step action sequences discovered by CEM across all 22 benchmark scenes. Inspecting the discovered sequences reveals several clear trends. First, CEM frequently schedules unconditional steps ($--$) in intermediate or late rollout stages, showing that the predictor can refine latent representations autonomously once initial geometric ambiguity has been resolved. Second, the selected features align closely with each scene's physical transport characteristics: planar environments such as *cornell-box* prioritize depth and geometric normal, whereas specular scenes such as *veach-mis* favor primitive index and shading normal. Finally, the rollout exhibits temporal feature staging, where later planning steps recruit slightly more AOVs, indicating that fine-grained geometric conditioning provides the greatest benefit after macroscopic light transport has stabilized.

E LIMITATIONS AND FUTURE WORK

While LeDenoiser demonstrates the viability of latent world models for rendering convergence, several open challenges remain:

Resolution and scalable encoding–decoding. Our experiments were conducted at a relatively modest resolution (336×336) primarily to enable efficient verification and rapid exploration of the latent dynamics framework. In order to better preserve fine rendering details, future iterations will need to scale to higher-resolution latent representations, which in turn calls for more efficient and scalable encoding and decoding architectures. Specifically, to alleviate the loss of compact, high-contrast features (such as the specular highlights and thin transmissive geometries observed in Section 4.2), promising directions include incorporating finer-grained sub-pixel output modules or introducing generative diffusion-based decoders, which hold strong potential to further elevate fine detail fidelity and denoising quality.

Amortized test-time planning. While CEM confirms that the discrete action space can be optimized at test time, population-based sampling adds substantial inference overhead. Future work could train learned value networks or distill the search policy into a feed-forward actor, enabling low-overhead AOV selection in interactive viewports.

Table 5: Per-step action plans discovered by CEM across all 22 scenes. Abbreviations: **A**lbedo, **D**epth, **G**eometric normal, **P**osition, **S**hading normal, **U**V, p**R**imitive index, s**H**ape index; $--$ denotes unconditional step ($K = 0$).

Scene	Step 0	Step 1	Step 2	Step 3	Step 4
coffee	S, U, R	U	D	P	–
cornell-box	U	D, G	D, G	U	A, D, G, P, S, R
dragon	P, U	P, S, U	A, D, S, R	G, P, U, H	D, R, H
glass-water	D, S	S	S	–	D, S, R
house	A, P, S, R	A, G, S, U, R	–	–	A, D, G, S
kitchen	A, P, H	H	–	–	A, G, P, S, U
lamp	A, P	A, R	–	–	–
lego	U	D, R	–	–	D, G, P, R
living-room	D, P, R, H	G	G	–	A, D, G, S, U
living-room-2	H	D, H	S	–	D, G, P, S, H
living-room-3	A	H	U, H	–	A, G, H
matpreview	D, U	D, G	G	–	P, S, H
rover	A, G, U, R	G, U, R	–	–	A, D, G, U
spaceship	A	–	G	–	–
staircase	A, D, G, H	A, H	–	–	D, S, H
staircase2	–	H	–	–	P, H
teapot	D, U, R, H	–	–	–	A, G, U
teapot-full	G, H	G, H	–	–	A, D, U
veach-ajar	D, G, U	D, G	A	U	D, G, H
veach-bidir	G, U	U	U	U	D, U, R, H
veach-mis	R	D	–	S	–
vol.-caustic	A	–	–	–	P, U

Cross-scene generalization. This work focuses on scene-specific light transport manifold fitting. Future research will explore multi-scene pretraining across large-scale synthetic rendering repositories, studying whether pre-trained latent dynamics can generalize zero-shot to unseen architectural scenes.