ITERATE, DON'T STACK: GROUNDED RECURSIVE TRANSFORMERS FOR GENERALIZABLE NOVEL VIEW SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing feed-forward novel view synthesis transformers apply a fixed amount of computation to render each target view per scene, although the complexity within a single scene is not distributed evenly. By fixing the number of transformer layers, they freeze the trade-off between quality and computation. We propose Universal View Synthesis Model, **UVSM**, capable to adaptively allocate computation within the target image. UVSM follows a unique principle: ground the scene once and recursively refine each token corresponding to image patches in parallel until convergence. A halting probability gate acting on refined tokens determines whether they need further refinement step. The halting gate acts as a dial at inference between a fast preview and a converged render. One trained model therefore replaces a family of fixed-depth models that a quality–compute curve would otherwise require. We further show that increasing the number of refinement steps in our network generalizes better than expanding the depth of a transformer network. Under the same training protocol, UVSM beats every baseline on all seven benchmarks with the highest margins on the out-of-distribution evaluations. The code and website are released as part of the supplementary material.

1 INTRODUCTION

Feed-forward novel view synthesis models render unseen target views per scene from a few posed input images in a single forward pass. The large transformers trained (Jin et al., 2025; Kim et al., 2026; Kang et al., 2026) on posed video sequences achieve the highest synthesis performance with minimal reliance on explicit 3D representations in contrast to prior state of the art (Charatan et al., 2024; Zhang et al., 2024). However they lock the amount of computation applied to a scene by the depth of the network chosen during training. On the other hand, the complexity within a single scene is not distributed evenly: an occlusion boundary, a thin railing, or a region visible in only one input view require deeper reasoning than a flat, well-observed wall, and a network of fixed depth applies the same computation to all of them. The same choice also fixes the trade-off between quality and computation, so that covering a range of operating points requires training a separate model for each.

UVSM resolves these constraints by grounding the scene once in a camera-invariant latent space via relative geometry embeddings, and then refines that latent space by iteratively applying a single weight-tied block. After every reasoning step, a halting probability estimator predicts whether the token has converged enough, adaptively controlling the depth allocated within an image, see Fig. 2, thus making the number of reasoning steps tunable at inference without retraining. As the refinement is shown to effectively converge towards a stationary solution, every intermediate state becomes a usable render, and the halting threshold becomes a dial between a fast preview and a converged result (Fig. 5). Additionally, we find that under the same matched protocol, increasing the number of recursive refinement steps of the tied block improves zero-shot transfer monotonically while increasing the number of distinct layers in the backbone leaves out-of-distribution quality flat or declining. The two axis to set depth of network remain distinguishable when we equalize the number of effective transformer layers applied per forward pass.

Claims and Contributions. UVSM establishes three claims. First, the reasoning depth in a feed-forward view synthesis networks can be allocated per patch and set at inference time using a weight-

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.

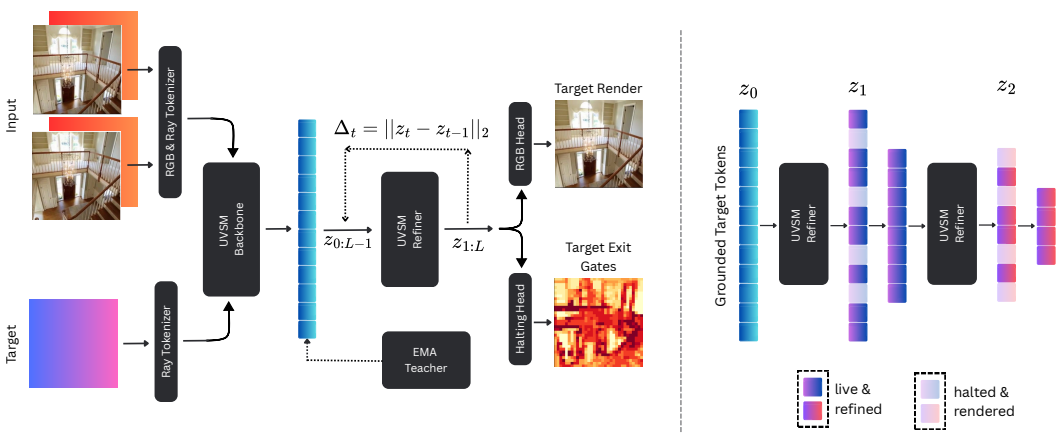


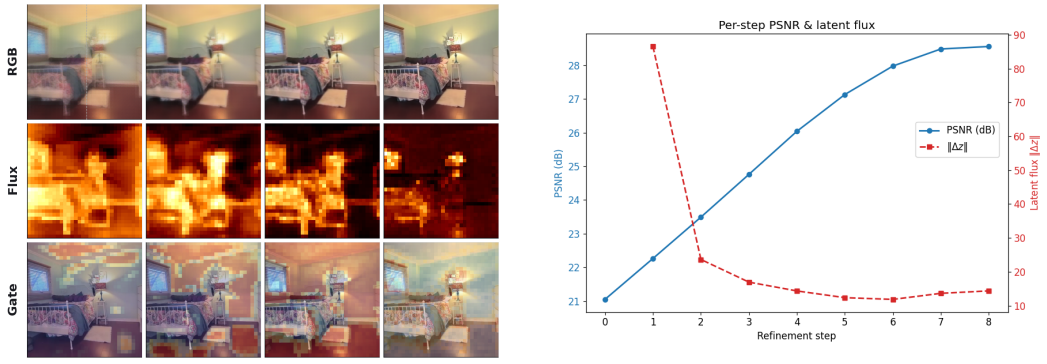
Figure 1: *Left - Architecture.* Input views are tokenised from RGB and Plücker rays, target views from rays alone; the Backbone maps the joint sequence to z_0 in one pass, and the UVSM Refiner with a tied set of weights is applied recursively. Every z_t is decoded through the RGB head to color space, and the shared Halting Gate scores the convergence probability per patch. The EMA Teacher exists only at training time for backbone distillation, and we define the latent flux as $\Delta_t = \|z_t - z_{t-1}\|_2$ (Fig. 2b). *Right - Efficient inference.* Target tokens grounded by the backbone are refined in parallel, and every patch is *halted* and leaves the sequence once its exit probability passes the test-time-tunable threshold, so the remaining steps are applied only to the tokens still *alive*.

tied recurrence, Secs. 3 and 7. Second, under a matched protocol, recurrent depth generalizes better than stacking a number of distinct transformer blocks, and the distinction survives when equalizing the compute, Sec. 6. Third, such an adaptive model leads seven retrained baselines on all seven benchmarks under standard PSNR, SSIM, LPIPS metrics, with its largest margins on zero-shot transfer, Sec. 5.

2 Related Work

Feed-forward novel view synthesis. The novel view synthesis architectures have recently transitioned from geometry-bound pipelines (Hong et al., 2024; Charatan et al., 2024; Chen et al., 2024; Zhang et al., 2024) to geometry-free transformers when LVSM (Jin et al., 2025) showed that an encoder–decoder and decoder-only transformers can render novel views with minimal 3D bias while outperforming all baselines without per-scene optimisation. SVSM (Kim et al., 2026) studies how such models scale, identifying the batch×target-view product as the effective batch size–the quantity our matched training protocol fixes, see Sec. 4, while RayZer (Jiang et al., 2025) removes pose supervision completely. All of these methods allocate a *fixed* amount of computation per view, although the complexity between views and even within a view can change non-uniformly. UVSM keeps the feed-forward, geometry-free formulation but replaces most of the deep transformer layers with one weight-tied block applied recurrently, so the parameter count is constant in the number of applications and the whole curve is traced from a single checkpoint.

Recurrent transformers. Reusing one block across depth and learning how long to continue the iterative evaluation goes back to the Universal Transformer (Dehghani et al., 2019), equilibrium models (Bai et al., 2019), ACT (Graves, 2017) and PonderNet (Banino et al., 2021). UVSM brings this family to novel view synthesis with two main differences. Halting is calculated *per patch*, and the exit distribution is held near uniform, which keeps intermediate renders usable and turns early exit into a working test-time dial, Sec. 7. Within NVS, iLRM (Kang et al., 2026) proposes a standard transformer pipeline in which low-resolution viewpoint tokens are refined layer by layer. That refinement, however, runs through *distinct* layers while the number of parameters grows with depth; the step count is fixed at training time. Our experiments show that such refinement built from the standalone layer saturates out-of-distribution performance, see Sec. 6. UVSM re-applies one weight-tied block instead, holding the parameter count constant and leaving the number of refinement steps tunable at inference time.



(a) refinement steps →

(b) zero-shot Scaling curves on ScanNet

Figure 2: UVSM in action. *(a)* The rendered target view, the latent flux of Fig. 1 and Halting Gate heatmaps at successive refinement steps, from left to right, across rows. Halting Gate stops refining walls while allocating more compute on footboard slats and lamp. *(b)* Per-step PSNR on zero-shot ScanNet plotted with the latent flux between consecutive steps. The flux falls to a small residual as quality reaches its full-depth value.

Deterministic refinement. UVSM iterates and trades computation for quality differently when compared against diffusion models. A diffusion model (Asim et al., 2026; Chan et al., 2023) is a *stochastic sampler* that reverses a noising process to draw one sample from a distribution (Rombach et al., 2022), so additional steps yield a different, better-distributed sample. UVSM is a *deterministic solver* with a weight-tied operator (Zhu et al., 2026; Geiping et al., 2025) that is iterated toward a stationary latent of a more refined reconstruction. The refinement in latent space directly improves all three metrics together until the latent features stop changing, Sec. 7. Determinism here is a design choice made early since a noisy refiner degraded quality considerably. This also separates UVSM from equilibrium models (Bai et al., 2019) with a fixed-point condition.

Camera conditioning. Ray-based conditioning has moved from concatenated Plücker coordinates (Sitzmann et al., 2022) to relative rotary forms with PRoPE (Li et al., 2025) rotating queries and keys by a ray-derived rotation, extending RoPE (Su et al., 2023b) so attention logits measure world-space proximity invariant to camera re-parameterisation. It is normally studied inside fixed-depth stacks; we examine how its effect depends on depth, see Sec. 8. Our EMA Teacher follows the self-supervision patterns (Grill et al., 2020; Caron et al., 2021; Assran et al., 2023; 2025) and is applied during training to anchor geometric latents of student backbone.

3 METHOD

UVSM has a unique principle: *ground the scene once, recursively refine each token of a target view in parallel until convergence* .

Grounding with UVSM Backbone. A pixel with ray origin o and direction d is represented with the Plücker coordinate $\gamma(o, d) = [\, o \times d \,\|\, d \,] \in \mathcal{R}^6$ and 3 RGB color channels; input-view patches concatenate RGB maps with the ray maps, target patches carry the ray maps alone. Non-overlapping input and target patches of size $h \times w$ are embedded, $\mathbb{R}^{h\times w\times 9} \to \mathbb{R}^D$ and $\mathbb{R}^{h\times w\times 6} \to \mathbb{R}^D$, using separate MLP tokenizers (Dosovitskiy et al., 2021) and processed as one *joint* sequence of shape $(V_{\text{in}}{+}V_{\text{tgt}},\ S,\ D)$, with S tokens per view. A 12-layer Backbone maps that sequence to the grounded latent z_0 in a single pass. The grounding is performed geometrically using alternating cross and self attention layers (Vaswani et al., 2023). Cross-attention draws keys and values from *input-view tokens only*, and PRoPE (Li et al., 2025) rotates Q and K by a rotation derived from each token's ray, measuring proximity in *world* space. Per-view self-attention with 2-D axial RoPE (Su et al., 2023a) adds intra-view layout, and a SwiGLU feed-forward network closes each block. An EMA Teacher (training-time only) observes the target RGB maps and predicts an oracle latent features z_0^T, anchoring the target frame tokens the recurrence iterates against.

UVSM Recursive Refiner. The Refiner is a 2-layer block of the same design, applied $L_{\max}{=}8$ times with *tied weights*, $z_t = \text{Refiner}(z_{t-1})\ \forall t \in \{1...L_{\max}\}$. The latent z_0 is itself the first reasoning

step, and every z_t decodes to a render $\hat{I}_t$. UVSM holds $12{+}2{=}14$ unique transformer blocks while a full-depth pass performs 12 grounding and $8{\times}2 = 16$ refinement steps that can be adaptively tuned during test time.

A single linear head with sigmoid non-linearity, shared across refinement steps, emits a per-patch exit probability on target tokens. We use a geometric distribution to turn exit probabilities into valid per-patch halting probabilities,

$$p(t) = \lambda_t \, S_{t-1}, \qquad S_t = S_{t-1}(1 - \lambda_t), \qquad S_0 = 1, \tag{1}$$

with the last step absorbing the remaining mass such that $\sum_t p(t) = 1$; $\mathbb{E}[t]$ is a per-patch average convergence depth under the halting distribution shown in bottom row of Fig. 5. While headline renders run full depth, we also report CDF-based early-exit performance (halt when $\sum_{s \le t} p(s) \ge \tau$), Sec. 7.

Training objective. Every render, backbone $\hat{I}_0$, intermediate renders, and final $\hat{I}_{L_{\max}}$ flows through the following cost function:

$$\mathcal{L} = \lambda_{\ell_2} \sum_{t=0}^{L_{\max}} p(t)\, \mathrm{MSE}_{\mathrm{pt}}(\hat{I}_t, I^*) + \lambda_{\mathrm{perc}} \mathcal{L}_{\mathrm{perc}}(\hat{I}_{L_{\max}}, I^*) + \lambda_{\mathrm{dist}} \|\mathbf{z}_0 - \mathbf{z}_0^T\|^2 + \lambda_{\mathrm{KL}} \mathrm{KL}\big(p \| \mathcal{U}\big), \tag{2}$$

where $\mathrm{MSE}_{\mathrm{pt}}$ is the per-patch MSE running over all $L_{\max}{+}1$ renders, $\mathcal{U}$ is a uniform prior over the halting probability distribution per target patch, $\mathbf{z}_0^T$ is a vector of teacher backbone tokens with access to target patches during training, and $\lambda_{\ell_2}{=}1.0$, $\lambda_{\mathrm{perc}}{=}0.2$, $\lambda_{\mathrm{dist}}{=}0.1$, $\lambda_{\mathrm{KL}}{=}0.01$. Three properties matter. *(i)* Gating applies to the MSE loss over patches; the perceptual term supervises the final render only, without gating. *(ii)* The KL term holds the halting probabilities p near uniform per target patch, preventing gate collapse so that multi-depth reasoning develops, Sec. 7. *(iii)* Backbone and the refiner are trained jointly from scratch, App. A.

Geometric Locking. PRoPE and the Teacher distillation are used to anchor a physically grounded, camera-invariant latent manifold. A PRoPE-grounded z_0 carries initial latent features per target view, and the Teacher, which sees the target as well as the input RGB, supplies an oracle latent that is geometrically correct by construction and pulls the Student's z_0 toward it. The weight-tied UVSM Refiner then operates in that grounded space at *every* step, so it can resolve correspondence ambiguity by attending along the physically correct epipolar structure, see Fig. 7, rather than converging to an attractor specific to the training distribution.

4 Experimental Setup

Baselines. We retrain every baseline method on the training subsets of Real Estate 10K and ACID, at the same effective batch size, defined as the batch×target-view product (Kim et al., 2026), and for the same number of iterations, following each method's author-provided architecture configuration. We re-evaluate all under one harness: SVSM, iLRM, LVSM encoder–decoder, the decoder-only LVSM at 18 and 24 layers, and both decoder-only depths with PRoPE attached. The +PRoPE variants serve as controls, pairing the strongest baselines with relative camera conditioning. Table 1 presents the 24-layer decoder-only model as *LVSM-dec-only*; the 18-layer pair is in App. C. DL3DV evaluations follow the original *benchmark* protocol for every method. As every method is retrained under a unified budget, absolute scores can vary in comparison to the original papers.

Benchmarks. *In-distribution* evaluations have been carried out using test subsets of RealEstate10K (Zhou et al., 2018) and ACID (Liu et al., 2021) at 2 input / 3 target views while *large-context* refers to the same datasets at 6/12; *Zero-shot* evaluations include ScanNet v2 (Dai et al., 2017) and DL3DV (Ling et al., 2023) on even and random target splits at 16/8. Large context benchmarks probe performance under increased context length and ScanNet and DL3DV probe generalization to new domains, overall creating a solid out-of-distribution evaluation suite for Table 1.

Metrics. We report PSNR, SSIM (Wang et al., 2004) and LPIPS (Zhang et al., 2018) (VGG) as the *mean over per-image scores* using the LVSM convention, at the full-depth final render; for early exit metrics see Sec. 7 and Fig. 5, every point with its realized depth.

Ablations. We have designed a comprehensive ablation protocol to understand the effect of individual components, namely, Backbone, Refiner, KL-regularization, Teacher distillation, and relative

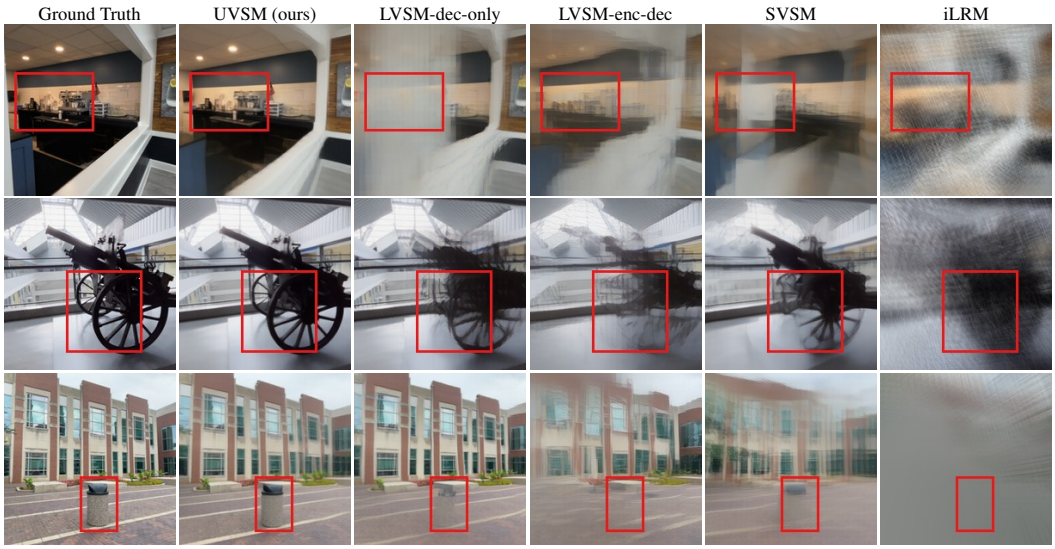


Figure 3: Zero-shot transfer results with one target view per row. The baselines smear the interior fittings, the gun carriage and the street trash bin into doubled or semi-transparent surfaces, and iLRM loses the scene entirely; UVSM locks the geometry once and focuses compute on complex textured patches. For more ScanNet and DL3DV results, see App. E.

Table 1: Quantitative evaluations; every method retrained under matched protocol. UVSM leads every benchmark, and the margins are largest out of distribution. Full baseline table is available under App. C. **Bold** = best, underline = second best.

	In-distribution (2/3)		Large-context (6/12)		Zero-shot (16/8)		
	RE10K	ACID	RE10K-L	ACID-L	ScanNet	DL3DV-E	DL3DV-R
PSNR (dB) ↑							
SVSM	27.27	28.78	20.07	23.09	19.98	16.25	16.27
iLRM	23.69	25.53	18.62	20.94	18.63	15.08	15.03
LVSM-enc-dec	26.50	27.62	20.55	22.49	21.26	16.17	16.12
LVSM-dec-only	28.14	29.76	24.33	25.89	25.27	19.76	19.66
LVSM-dec-only+PRoPE	28.50	30.12	22.88	24.54	24.33	18.79	18.72
UVSM (ours)	**28.71**	**30.16**	**25.84**	**26.95**	**28.56**	**21.41**	**21.25**
SSIM ↑							
SVSM	0.868	0.837	0.648	0.649	0.644	0.382	0.379
iLRM	0.752	0.714	0.628	0.615	0.622	0.390	0.388
LVSM-enc-dec	0.849	0.801	0.662	0.618	0.665	0.367	0.366
LVSM-dec-only	0.882	0.859	0.796	0.755	0.813	0.572	0.569
LVSM-dec-only+PRoPE	0.887	0.863	0.741	0.703	0.748	0.519	0.517
UVSM (ours)	**0.892**	**0.865**	**0.821**	**0.780**	**0.841**	**0.640**	**0.639**
LPIPS ↓							
SVSM	0.128	0.158	0.314	0.315	0.437	0.517	0.515
iLRM	0.338	0.365	0.448	0.460	0.586	0.686	0.684
LVSM-enc-dec	0.156	0.208	0.308	0.335	0.419	0.556	0.553
LVSM-dec-only	0.117	0.137	0.195	0.219	0.250	0.347	0.346
LVSM-dec-only+PRoPE	0.112	0.132	0.221	0.242	0.269	0.357	0.355
UVSM (ours)	**0.108**	**0.129**	**0.178**	**0.199**	**0.193**	**0.295**	**0.293**

camera conditioning. All ablation experiments share the same learning protocol with training details under App. A, and differ only in the ablated axis. Some ablations use physically shared runs.

5 Main Results

Table 1 presents performance of UVSM along with the baselines, all retrained under a unified protocol. UVSM leads every baseline in each benchmark with the largest margins on five out-of-distribution settings. On ScanNet, we improve on the best baseline by 3.3 dB.

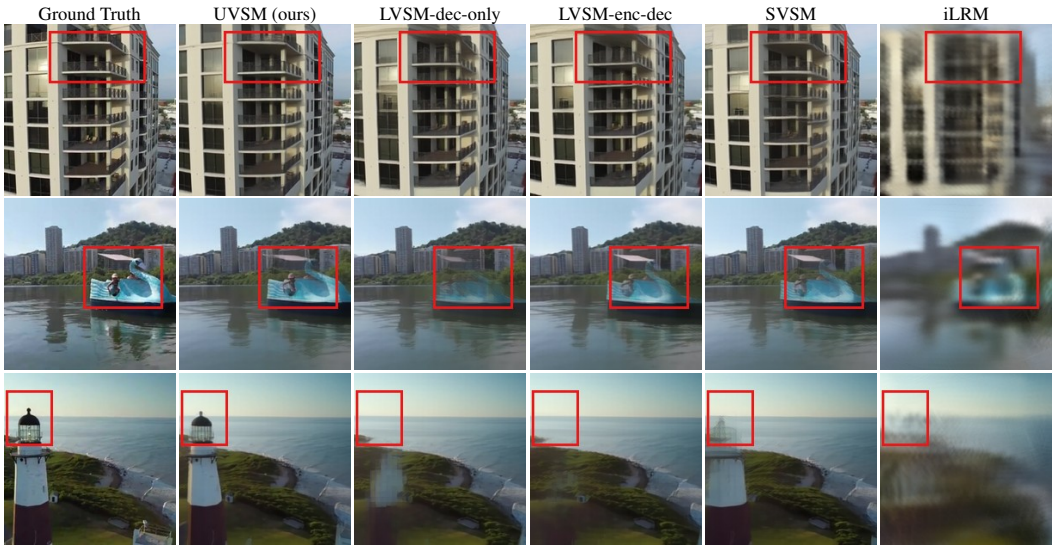


Figure 4: In-distribution scenes, same three metrics and protocol at 2 input / 3 target views. Top balcony railings, the pedal boat and the lighthouse lantern are where the methods differ; the differences are small, consistent with the narrow spread in Table 1. All target views are in App. E.

Quantitative profile. The quantitative performance of UVSM profiles as the recurrent design predicted. Following Sec. 3, the recurrence contributes by resolving correspondence inside a grounded latent with small gains where correspondence is easy to lock and large gains where it is hard. In-distribution test scenes come from the distribution the model was fitted on, and correspondence is largely settled by the first pass. The residual error is then dominated by appearance details the input views do not carry. For in-distribution scenes, the leading methods sit within a fraction of a decibel of one another, and our margin over the strongest is correspondingly small. Out of distribution, under camera configurations and scene statistics not seen in training, correspondence between target and the input view patches is uncertain and further refinement can revise it, so the margin opens by as large as 3 dB decibels in favor of UVSM.

Qualitative profile. The qualitative comparisons are informative, especially out-of-distribution, as the baselines do not degrade gracefully, with many failing structurally; see Fig. 3. Since they resolve the correspondence inconsistently, surfaces appear doubled or semi-transparent with blurry visual artifacts across views. UVSM places the structure correctly and refines the complex patches iteratively until convergence with little residual softness on texture. In distribution, Fig. 4, the ordering persists, but the differences shrink to fine detail as the narrow numerical band of quantitative experiments implies.

Parameter count. UVSM holds fewer unique transformer parameters than the decoder-only LVSM baseline, see Table 5; yet leads with the quantitative and qualitative advantages rising with the number of recursive refinement iterations; see the matched control of Sec. 6. We also find that the camera-conditioned decoder-only LVSM is the strongest baseline on in-distribution samples yet falls behind its own unconditioned variant on every out-of-distribution benchmark, see Table 1.

6 Recurrent Depth as a Generalization Mechanism

We have built an extensive ablation stack around UVSM to distill the contribution of each component fairly and they share a common training protocol while moving one knob each. Table 2 isolates components at a fixed backbone, and two sweeps vary the number of *distinct* backbone layers and the number of *recursive refinement* steps of the tied blocks separately, see Table 6, App. D.

Component contributions. Recurrence is the largest single contribution in Table 2, and its gain is an order of magnitude larger for out-of-distribution than in-distribution performance, following the reasoning outlined in Sec. 3 and Sec. 5. The Refiner's work is resolving correspondence inside a grounded latent manifold (Fig. 7), which becomes more difficult in a new domain requiring

Table 2: Component ablation at a fixed 12-layer backbone. Rows are additive in contributions from a1 to a4, and a5 removes PRoPE from the full model. Recurrence (a1→a2) is the largest single contribution and carries the out-of-distribution gain. **Bold** = best, underline = second best.

	In-distribution (2/3)		Large-context (6/12)		Zero-shot (16/8)		
	RE10K	ACID	RE10K-L	ACID-L	ScanNet	DL3DV-E	DL3DV-R
PSNR (dB) ↑							
a1 Backbone only ($L_{\max}$=0)	28.45	29.99	23.90	25.51	25.96	19.38	19.24
a2 + recurrence, no KL	28.63	30.14	25.69	**26.98**	28.45	21.16	20.91
a3 + KL gate prior	28.60	30.12	25.51	26.81	28.36	20.95	20.76
a4 + distillation (full UVSM)	**28.71**	**30.16**	**25.84**	26.95	**28.56**	**21.41**	**21.25**
a5 full, PRoPE off	28.36	29.85	24.94	26.16	28.43	21.00	20.84
SSIM ↑							
a1 Backbone only ($L_{\max}$=0)	0.886	0.862	0.770	0.733	0.781	0.543	0.538
a2 + recurrence, no KL	0.890	0.864	0.818	0.780	0.838	0.629	0.623
a3 + KL gate prior	0.890	0.864	0.812	0.774	0.835	0.623	0.617
a4 + distillation (full UVSM)	**0.892**	**0.865**	**0.821**	**0.780**	0.841	**0.640**	**0.639**
a5 full, PRoPE off	0.887	0.860	0.796	0.753	**0.844**	0.624	0.621
LPIPS ↓							
a1 Backbone only ($L_{\max}$=0)	0.114	0.134	0.219	0.241	0.254	0.358	0.357
a2 + recurrence, no KL	0.111	0.130	0.179	**0.197**	0.193	**0.292**	**0.293**
a3 + KL gate prior	0.109	0.130	0.182	0.200	**0.191**	0.297	0.297
a4 + distillation (full UVSM)	**0.108**	**0.129**	**0.178**	0.199	0.193	0.295	0.293
a5 full, PRoPE off	0.113	0.136	0.198	0.224	0.198	0.300	0.301

extra recursions. Teacher distillation anchors the feature manifold, improving the latent space the refinement iterations start from and therefore the rendered output they converge to, leading to metric gains in the converged results. Camera conditioning supplies the relative inter-frame information the iteration exploits, and removing it from the full model results in a very small quality decrease. Following Sec. 5, we nevertheless report the best LVSM-decoder-only baseline with PRoPE to keep a fair comparison with UVSM. Although the gate prior is endpoint-neutral, we present scaling curves, see Fig. 6 under App. A.1, with a clear advantage of the KL-regularized recursive refinement process beating the no-KL version by 5.7 dB mid-depth. Thus, UVSM can allocate the reasoning depth in a feed-forward rendering process using a convergent weight-tied recurrence, verifying the first claim of the paper (Sec. 1).

Increasing depth vs increasing refinment iteration. Stacking backbone layers improves in-distribution quality slightly and monotonically, and then stops for out-of-distribution cases, where quality peaks at an intermediate depth and falls again. In contrast, increasing the number of recursive iterations of the tied block improves every zero-shot benchmark monotonically on all three metrics while leaving the parameter count untouched. The two knobs therefore move out-of-distribution quality in opposite directions, which was the second claim and central observation of UVSM (Sec. 1), see Table 6 under App. D. Specifically, the $b3$ configuration stacks sixteen distinct backbone layers and applies the tied block eight times while the $c3$ configuration stacks eight distinct backbone layers and recursively refines twelve times. Although both configurations perform the same number of layer applications per forward pass, the recurrent allocation ($c3$) wins every zero-shot benchmark.

Corroboration and Scope. The same saturation appears inside the LVSM baseline family. Expanding the decoder-only model from eighteen to twenty-four layers yields a small in-distribution gain and a modest zero-shot boost that leaves it far short of UVSM (App. C). Standalone-depth saturation is therefore a property of the regime rather than of our backbone.

7 Adaptive Per-Token Compute

The benefit of recurrence appears across refiner steps making UVSM's depth tunable at inference.

Convergence of the refinement. Refinement is monotone in all three metrics and the magnitude of change in the latent manifold between steps contracts toward a small residual (Fig. 5, top; Fig. 2b), which together guarantee early exit. Monotonicity implies an intermediate render is a slightly blurry version of the final answer rather than a different one, so stopping early costs quality without changing what is rendered. Contraction means the remaining steps have progressively less to change, so the cost of stopping falls as depth increases. An iterative model lacking either property would still

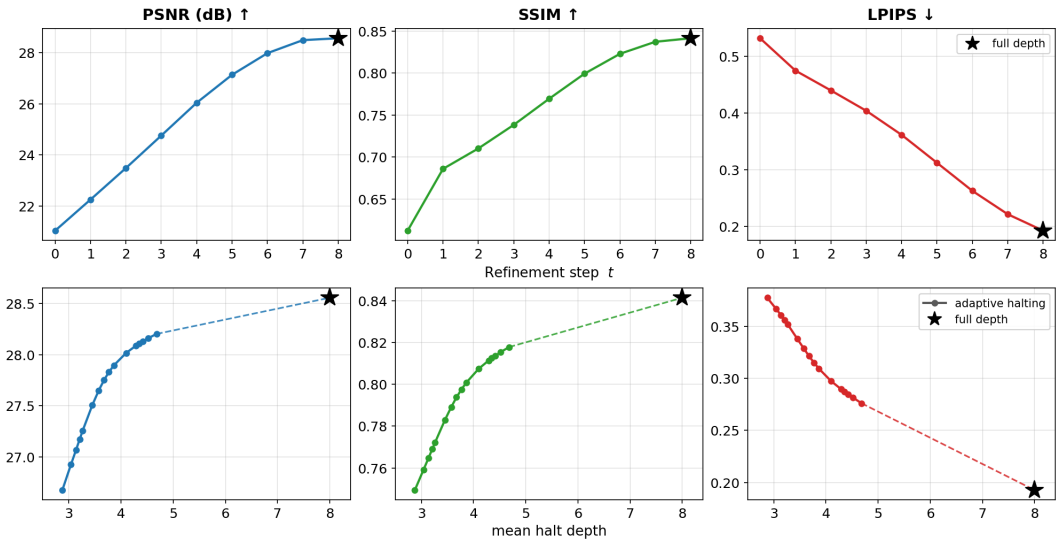


Figure 5: Per-step quality and compute behaviour of one trained model on zero-shot ScanNet. *Top*: PSNR, SSIM and LPIPS of the rendered output produced after each refinement step t, aggregated over the evaluation set, with the full-depth values marked. *Bottom*: the same three metrics plotted against the mean realized recursion depth as the per-patch halting threshold is swept over $[0.02, 0.98]$ during inference-time parallel decoding; the star marks full depth and the tabulated values are given in App. D. Quality improves at every step and most of the full-depth quality is reached at roughly half the mean depth.

improve with depth but would be unusable as a dial, because its intermediate states would not be renders one could ship, see no-KL version in Fig. 6. The three metrics move together throughout, differently from the trade-off a diffusion sampler makes between fidelity and perceptual quality. The latent settles to a small non-zero residual rather than to a fixed point, so we describe the behavior as convergence to a plateau rather than as an equilibrium.

Per-patch halting. Because the exit distribution is computed and normalised per patch, and target patches decode independently, the remaining refinement steps are applied only to the tokens still alive, see Fig. 1. Sweeping the halting threshold therefore traces quality against the *mean* realized depth actually spent (Fig. 5, bottom). The allocation behind that mean is not uniform: the expected-exit-step map follows the complexity of scene structure (Fig. 1), and adaptive halting never exhausts the budget even at the most conservative threshold, since easy regions stop as early as possible, proving the UVSM's argument for adaptive allocation. A fixed-depth model spends its depth uniformly with the same computation allocated even for occlusion boundaries and thin structures, which need it most, whereas per-patch halting removes it from the regions that do not. At equal mean depth the two allocations are therefore not equivalent, and the curve we measure is the one a fixed-depth family cannot reach by choosing a smaller stack. The scaling curve in Fig. 2 itself is steep and then flat: most of the attainable quality arrives at a compute little under half the full depth, and the remainder buys a further gain in fidelity.

An unregularised gate collapses. Since the standalone reconstruction loss doesn't reward the network for releasing a token early, the network can easily collapse to a full recursive or backbone-alone forward pass. The halting gate can minimise reconstruction loss either by exiting trivially at the first step or by running every patch to full depth as the ablation without the KL prior does. In both cases the exit distribution degenerates and only one render is effectively supervised. Holding the batch-averaged exit distribution near uniform removes those degenerate optima by placing weight on every step, so a multi-depth refinement strategy develops before the gate is free to specialize, and each intermediate render is optimized as an output independently. The refinement trajectory in the latent manifold is transformed, so quality arrives early, and every depth is usable, instead of most of the gain landing in the final steps (App. A.1). Thus, the KL-prior creates an internal mechanism for a usable quality–compute curve, making the halting sweep meaningful.

8 DISCUSSION

Generalization mechanism. Our extensive ablation suite has already settled the evidence for UVSM's advantage and how it is rising with recursive computation: $b3$ and $c3$ apply the same number of effective transformer layers, and the recurrent allocation still wins every zero-shot benchmark. The mechanism we proposed in Sec. 3 with a single grounding stage followed by recursive refinement works as intended. The iteration settles rather than drifts, and the cross-attention probes of App. B show queries that spread their mass along broad, elongated bands at the backbone stage producing compact responses on the corresponding input regions after refinement. Removing camera conditioning slightly degrades quality, mostly on the wide-baseline settings.

Adaptive computation. UVSM makes depth a tunable inference-time knob because refinement is monotone and the latent flux contracts, so an intermediate render is a degraded version of the final answer rather than a different state. Remaining steps are applied only to tokens still alive, and the expected-exit map follows scene complexity structure instead of spreading depth evenly. The gate prior keeps the regime reachable, since an unregularized gate degenerates to a single supervised render.

Camera conditioning. Ray-relative conditioning is normally attached to fixed-depth transformer stacks and reported as an in-distribution gain, which our experiments reproduce at both depths we test. During out-of-distribution experiments, the same intervention degrades every benchmark on every metric, while inside UVSM it helps nearly everywhere. A fixed-depth model commits to an association between target rays and input regions in a single pass, whereas a recurrence re-enters the grounded space at every step and can correct an initial mis-association.

Implications and future directions. With UVSM design, we have shown that scaling up the renderer beyond its training distribution can be achieved by iterations of a grounded operator rather than by additional distinct layers. We leave one question open for future research. Our evidence comes from one matched protocol at a single resolution, so the experiments and conclusions are internally controlled. We believe that pre-training under large and more varied in-the-wild data can boost the performance of UVSM even further, opening up new research directions for efficient training and inference techniques using adaptive compute allocation.

9 CONCLUSION

The feed-forward computation of transformer-based view synthesis models can be adaptively tuned towards capturing the scene complexity. We have shown that repeatedly applying one grounded, weight-tied refinement block converges to a stationary solution, and that a per-patch halting gate can allocate depth within an image at inference. Under a matched protocol recursive refinement steps, rather than the number of distinct layers, preserve its performance under distribution shift, and the attention probes show the refinement resolving correspondence as the mechanism predicts. We plan to analyze UVSM's potential for efficient pre-training and inference using adaptive computation in the future.

AI USE STATEMENT

We wrote and revised the manuscript while using generative AI assistance for grammar editing and typesetting the paper's tables and figures from the project's experimental record. We did not use generative AI to design the model, run or analyse the experiments, or produce any experimental result. We verified every claim, table and figure against the underlying artifacts and take responsibility for the final content of this work.

ETHICS STATEMENT

This work uses standard, publicly released research datasets (RealEstate10K, ACID, ScanNet, DL3DV) under their respective licenses and involves no human-subject data collection.

Reproducibility statement

The original uvsm codebase and project website are included in the supplementary material, together with the full training and evaluation protocol of App. A; trained UVSM checkpoints will be released upon acceptance.

References

Mohammad Asim, Christopher Wewer, and Jan Eric Lenssen. Scenetok: A compressed, diffusable token space for 3d scenes, 2026. URL https://arxiv.org/abs/2602.18882.

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, 2023. URL https://arxiv.org/abs/2301.08243.

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, 2025. URL https://arxiv.org/abs/2506.09985.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models, 2019. URL https://arxiv.org/abs/1909.01377.

Andrea Banino, Jan Balaguer, and Charles Blundell. Pondernet: Learning to ponder, 2021. URL https://arxiv.org/abs/2107.05407.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. URL https://arxiv.org/abs/2104.14294.

Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models, 2023. URL https://arxiv.org/abs/2304.02602.

David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction, 2024. URL https://arxiv.org/abs/2312.12337.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. *MVSplat: Efficient 3D Gaussian Splatting from Sparse Multi-view Images*, pp. 370–386. Springer Nature Switzerland, October 2024. ISBN 9783031726644. doi: 10.1007/978-3-031-72664-4_21. URL http://dx.doi.org/10.1007/978-3-031-72664-4_21.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes, 2017. URL https://arxiv.org/abs/1702.04405.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers, 2019. URL https://arxiv.org/abs/1807.03819.

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, 2021. URL https://arxiv.org/abs/2010.11929.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach, 2025. URL https://arxiv.org/abs/2502.05171.

Alex Graves. Adaptive computation time for recurrent neural networks, 2017. URL https://arxiv.org/abs/1603.08983.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. URL https://arxiv.org/abs/2006.07733.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d, 2024. URL https://arxiv.org/abs/2311.04400.

Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, and Georgios Pavlakos. Rayzer: A self-supervised large view synthesis model, 2025. URL https://arxiv.org/abs/2505.00702.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias, 2025. URL https://arxiv.org/abs/2410.17242.

Gyeongjin Kang, Seungtae Nam, Seungkwon Yang, Xiangyu Sun, Sameh Khamis, Abdelrahman Mohamed, and Eunbyung Park. ilrm: An iterative large 3d reconstruction model, 2026. URL https://arxiv.org/abs/2507.23277.

Evan Kim, Hyunwoo Ryu, Thomas W. Mitchel, and Vincent Sitzmann. Scaling view synthesis transformers, 2026. URL https://arxiv.org/abs/2602.21341.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding, 2025. URL https://arxiv.org/abs/2507.10496.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision, 2023. URL https://arxiv.org/abs/2312.16256.

Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image, 2021. URL https://arxiv.org/abs/2012.09855.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Noam Shazeer. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering, 2022. URL https://arxiv.org/abs/2106.02634.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023a. URL https://arxiv.org/abs/2104.09864.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023b. URL https://arxiv.org/abs/2104.09864.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.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.

Biao Zhang and Rico Sennrich. Root mean square layer normalization, 2019. URL https://arxiv.org/abs/1910.07467.

Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. GS-LRM: Large reconstruction model for 3D Gaussian splatting. In *European Conference on Computer Vision (ECCV)*, 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images, 2018. URL https://arxiv.org/abs/1805.09817.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, Ziniu Li, Haoran Que, Boyi Wei, Zixin Wen, Fan Yin, He Xing, Lu Li, Jiajun Shi, Kaijing Ma, Shanda Li, Taylor Kergan, Andrew Smith, Xingwei Qu, Mude Hui, Bohong Wu, Qiyang Min, Hongzhi Huang, Xun Zhou, Wei Ye, Jiaheng Liu, Jian Yang, Yunfeng Shi, Chenghua Lin, Enduo Zhao, Tianle Cai, Ge Zhang, Wenhao Huang, Yoshua Bengio, and Jason Eshraghian. Scaling latent reasoning via looped language models, 2026. URL https://arxiv.org/abs/2510.25741.

A TRAINING DETAILS

Optimisation. AdamW (Loshchilov & Hutter, 2019) with β_1=0.9, β_2=0.98, weight decay 0.05; peak learning rate 3×10^{-4} with cosine decay, 8,192 warm-up steps, 100K total steps; bf16 autocast with TF32. Gradients are clipped at norm 1.6, and the optimiser step is *skipped* whenever the gradient norm exceeds $5\times$ the clip threshold. Gradient checkpointing is mandatory at this batch size: every second backbone block plus the full recurrence are checkpointed (the dense backbone alone is $\sim$97 GiB at per-GPU batch 8). Global batch is 24 (3 GPUs $\times$ 8). Training data is RealEstate10K and ACID training scenes with frame distances sampled in $[25, 192]$, 256^2 images (G=32, 1024 patch tokens per view), 2 input / 6 target views.

Joint training. There is no staged freezing: every forward pass runs the full Backbone→Refiner loop, every optimiser step updates Backbone, Refiner, and ExitGate simultaneously, and the Teacher EMA (α=0.999) is updated after every step—co-adaptation lets the Backbone learn representations amenable to refinement while the Refiner adapts to the Backbone's evolving geometry. The KL weight ramps linearly $0 \rightarrow 0.01$ over the warm-up steps, then stays constant. When $L_{\max}$=0 (the backbone-only ablation), halting is trivial and the single render receives full-weight MSE and perceptual supervision.

Stability design notes. Cross-attention uses per-head RMSNorm (Zhang & Sennrich, 2019) QK-normalisation before the PRoPE rotation; the per-view self-attention output projection is zero-initialised (exact identity at initialisation); FFNs are SwiGLU (Shazeer, 2020). Softmaxes over a single view's G^2=1024 tokens have lower entropy than cross-attention over the full VG^2 sequence, reducing gradient variance; together with QK-normalisation and raising the gradient clip from 0.6 to 1.6, these choices eliminated the training divergences of early runs. They are optimisation-stability choices—no performance claim rests on them, and per-view self-attention is held enabled across all ablation runs.

Inference and evaluation. Evaluation uses the same checkpoint throughout at 256^2 in bf16. Headline renders decode $\hat{I}_{L_{\max}}$ with every token run to full depth; the ExitGate is applied only in the halting sweep (Sec. 7), where it is an inference-time read-out of the survival CDF and changes no weights. The sweep uses 800 zero-shot ScanNet images and reports, for each τ, the mean realized depth over all target patches alongside the three metrics. Metrics are computed by a unified aggregation strategy for every method, as the mean over per-image scores.

Training-time monitoring. The manifold-convergence tracker $\varepsilon_m \leftarrow \beta\varepsilon_m + (1-\beta)\,\mathbb{E}\|z_0 - z_0^T\|$ with β=0.99 tracks the mean distillation distance; a declining ε_m indicates the student latent approaching the full-context oracle. We use it as a monitoring diagnostic only (Sec. 8).

A.1 THE KL GATE PRIOR: WHAT IT CHANGES ABOUT THE TRAJECTORY

Figure 6 and Table 3 give the complete per-step data behind Sec. 7: the a2 (no KL) and a3 (+KL) ablations on RealEstate10K *and* ACID, all three metrics at every step $t = 0 \ldots 8$. The final step coincides with each configuration's full-depth headline render. On ACID the halfway-step PSNR gap is +5.7 dB and the two endpoints there agree to ≤ 0.02 dB / 0.001 SSIM (the ≤ 0.03 dB bound quoted elsewhere is the looser of the two datasets).

The full model. The per-step trajectory of the released model on zero-shot ScanNet is Fig. 5 (top); its step-0 (backbone) render sits far below the backbone-only ablation a1 (25.96 dB, Table 2), because inside UVSM the Backbone is a grounding stage feeding the recurrence, not a standalone renderer trained to finish the image.

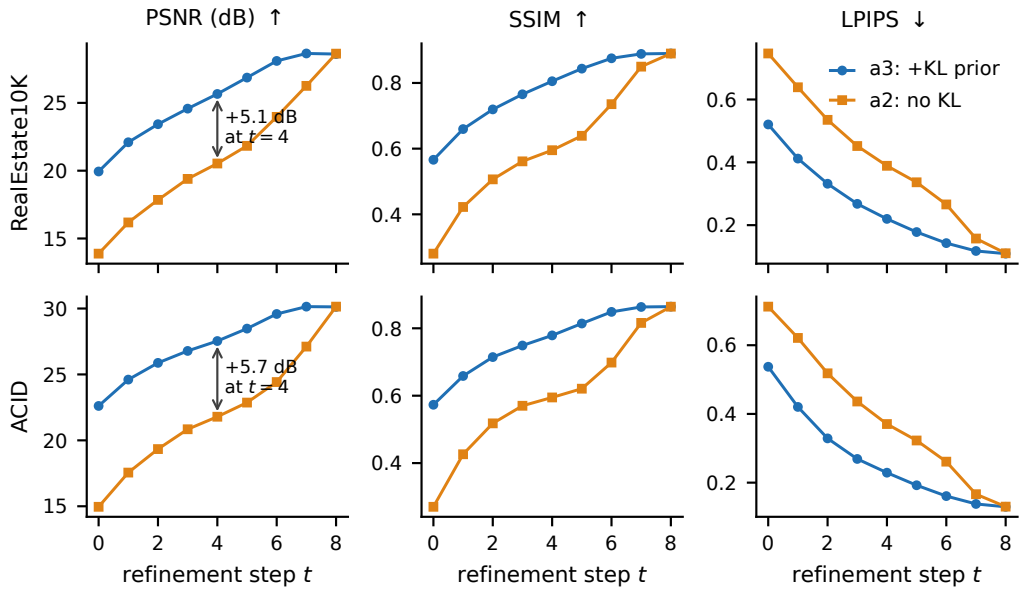


Figure 6: **Per-step trajectories on both datasets.** With the KL prior, quality is front-loaded and every intermediate render is usable; without it, useful quality arrives only in the final steps (32–38% of the total PSNR gain in the last two steps, vs. 6–7% with KL).

Table 3: **Per-step PSNR/SSIM/LPIPS, a2 (no KL) vs. a3 (+KL).** Endpoints agree to ≤ 0.03 dB PSNR / 0.001 SSIM.

step t	a2 (no KL)			a3 (+KL)		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
RealEstate10K (2/3)						
0	13.88	0.280	0.747	19.94	0.566	0.521
1	16.18	0.422	0.639	22.10	0.660	0.412
2	17.85	0.507	0.535	23.43	0.720	0.332
3	19.40	0.561	0.452	24.58	0.766	0.268
4	20.53	0.595	0.389	25.66	0.805	0.220
5	21.83	0.639	0.336	26.87	0.844	0.178
6	23.97	0.735	0.266	28.09	0.875	0.143
7	26.26	0.850	0.157	28.64	0.889	0.118
8	28.63	0.890	0.111	28.60	0.890	0.109
ACID (2/3)						
0	14.96	0.270	0.712	22.61	0.573	0.537
1	17.56	0.426	0.621	24.61	0.659	0.420
2	19.34	0.518	0.518	25.88	0.715	0.329
3	20.84	0.570	0.436	26.78	0.749	0.269
4	21.79	0.595	0.371	27.53	0.779	0.229
5	22.87	0.621	0.323	28.48	0.814	0.192
6	24.42	0.698	0.261	29.59	0.849	0.161
7	27.11	0.816	0.166	30.14	0.863	0.138
8	30.14	0.864	0.130	30.12	0.864	0.130

B CROSS-ATTENTION PROBES

Geometric Locking predicts that refinement should sharpen correspondence: a target query that is initially ambiguous about which input region explains it should concentrate its attention on the correct input patches as the tied block is re-applied,. Figure 7 inspects that directly. We select three query patches in the target view, marked in the left column, and visualise where each one places its cross-attention mass in the two input views, once at the Backbone pass that produces z_0 and once after refinement.



Figure 7: Cross-attention of three target queries, RealEstate10K, 2 input views. Left column: the target view with the three query patches marked. Centre and right: two input views, overlaid with the attention mass each query assigns to them. *Top row*: the 12-layer Backbone that produces z_0. *Bottom row*: the same queries after refinement by the weight-tied block.

At the Backbone stage the responses are broad and horizontally elongated, each query spreading its mass along a band rather than settling on a location. That is the shape an unresolved correspondence takes when the model has narrowed a query to a region of the scene but not yet to the point within it. After refinement the same three queries produce compact responses, positioned on the content the target patch corresponds to in each input view, and the elongation is largely gone.

C Full Baseline Table

Table 4 is Table 1 with the 18-layer decoder-only LVSM variants restored: seven retrained baselines, all seven benchmarks, all three metrics, one protocol, full depth. UVSM is best in all 21 columns here as well. The 18-layer pair matters twice: the 18-layer model is the strongest baseline on large-context RealEstate10K/ACID PSNR, and its +PRoPE contrast is the second depth at which fixed-depth camera conditioning is measured. Together with Table 2 (a4 vs. a5) this table carries every PRoPE contrast the paper makes: the fixed-depth effect is the difference between the ±PRoPE rows at each depth, and the recurrent effect is a4 vs. a5.

Table 4: **Full baseline comparison, seven retrained baselines.** Full depth ($L_{\max}$=8), DL3DV benchmark protocol. **Bold** = best, underline = second best per column.

	In-distribution (2/3)		Large-context (6/12)		Zero-shot (16/8)		
	RE10K	ACID	RE10K-L	ACID-L	ScanNet	DL3DV-E	DL3DV-R
PSNR (dB) ↑							
SVSM	27.27	28.78	20.07	23.09	19.98	16.25	16.27
iLRM	23.69	25.53	18.62	20.94	18.63	15.08	15.03
LVSM-enc-dec	26.50	27.62	20.55	22.49	21.26	16.17	16.12
LVSM-dec-only (18L)	27.97	29.64	24.34	25.95	24.48	19.15	19.00
LVSM-dec-only (18L)+PRoPE	28.30	29.97	21.70	23.38	22.69	17.57	17.52
LVSM-dec-only	28.14	29.76	24.33	25.89	25.27	19.76	19.66
LVSM-dec-only+PRoPE	28.50	30.12	22.88	24.54	24.33	18.79	18.72
UVSM (ours)	**28.71**	**30.16**	**25.84**	**26.95**	**28.56**	**21.41**	**21.25**
SSIM ↑							
SVSM	0.868	0.837	0.648	0.649	0.644	0.382	0.379
iLRM	0.752	0.714	0.628	0.615	0.622	0.390	0.388
LVSM-enc-dec	0.849	0.801	0.662	0.618	0.665	0.367	0.366
LVSM-dec-only (18L)	0.879	0.856	0.794	0.755	0.799	0.536	0.531
LVSM-dec-only (18L)+PRoPE	0.884	0.860	0.695	0.655	0.694	0.454	0.448
LVSM-dec-only	0.882	0.859	0.796	0.755	0.813	0.572	0.569
LVSM-dec-only+PRoPE	0.887	0.863	0.741	0.703	0.748	0.519	0.517
UVSM (ours)	**0.892**	**0.865**	**0.821**	**0.780**	**0.841**	**0.640**	**0.639**
LPIPS ↓							
SVSM	0.128	0.158	0.314	0.315	0.437	0.517	0.515
iLRM	0.338	0.365	0.448	0.460	0.586	0.686	0.684
LVSM-enc-dec	0.156	0.208	0.308	0.335	0.419	0.556	0.553
LVSM-dec-only (18L)	0.120	0.140	0.197	0.217	0.266	0.357	0.359
LVSM-dec-only (18L)+PRoPE	0.115	0.134	0.255	0.273	0.325	0.403	0.400
LVSM-dec-only	0.117	0.137	0.195	0.219	0.250	0.347	0.346
LVSM-dec-only+PRoPE	0.112	0.132	0.221	0.242	0.269	0.357	0.355
UVSM (ours)	**0.108**	**0.129**	**0.178**	**0.199**	**0.193**	**0.295**	**0.293**

Table 5 gives the parameter count of each model. UVSM has 22% fewer parameters than LVSM-dec-only while leading it on every benchmark, because depth in UVSM is obtained by re-applying a 2-layer block rather than by adding distinct layers. PRoPE applies a rotation derived from each ray and has no learned weights, so the +PRoPE variant has the same count as its base model.

Table 5: Parameter counts. Transformer layers lists distinct layers; UVSM's 2-layer Refiner is weight-tied and applied up to eight times, so its full-depth pass performs 28 layer applications with 14 distinct layers.

Model	Family	Transformer layers	Parameters
SVSM	encoder–decoder	8 + 8	127M
iLRM	encoder–decoder	24	171M
LVSM-enc-dec	encoder–decoder	12 + 12	173M
LVSM-dec-only	decoder-only	24	171M
LVSM-dec-only+PRoPE	decoder-only	24	171M
UVSM (ours)	encoder–decoder	12 + 2 tied (×8)	**133M**

D DEPTH SWEEPS AND THE HALTING-THRESHOLD SWEEP

Two questions are answered here: whether the two depth knobs move out-of-distribution quality in the same direction (Table 6), and what quality costs at each realized depth (Table 7).

Table 6: **The two depth knobs, swept separately** (Sec. 6). *(a)* Stacking backbone layers ($L_{\max}$=8 fixed) buys mild in-distribution gains, nothing consistent out of distribution. *(b)* Adding recurrent depth (8-layer backbone) improves *every* zero-shot benchmark monotonically on *all three* metrics; large context is flat-to-negative. Shared runs b2 = a4, c2 = b1. Full depth. **Bold** = best, underline = second best.

(a) Backbone-layer sweep (standalone-layer depth)

	In-distribution (2/3)		Large-context (6/12)		Zero-shot (16/8)		
	RE10K	ACID	RE10K-L	ACID-L	ScanNet	DL3DV-E	DL3DV-R
PSNR (dB) ↑							
b1 8 layers (= c2)	28.59	30.08	25.79	26.92	**28.60**	21.29	21.20
b2 12 layers (= a4, full)	28.71	30.16	**25.84**	26.95	28.56	**21.41**	**21.25**
b3 16 layers	**28.78**	**30.22**	25.73	**26.99**	28.49	21.23	21.13
SSIM ↑							
b1 8 layers (= c2)	0.890	0.863	0.820	0.780	**0.843**	0.636	0.636
b2 12 layers (= a4, full)	0.892	0.865	**0.821**	**0.780**	0.841	**0.640**	**0.639**
b3 16 layers	**0.893**	**0.866**	0.813	0.776	0.838	0.633	0.632
LPIPS ↓							
b1 8 layers (= c2)	0.110	0.131	0.180	0.201	0.194	0.308	0.306
b2 12 layers (= a4, full)	0.108	0.129	**0.178**	**0.199**	0.193	0.295	0.293
b3 16 layers	**0.107**	**0.129**	0.184	0.203	**0.192**	**0.291**	**0.288**

(b) Recursion-depth sweep (recurrent depth)

	In-distribution (2/3)		Large-context (6/12)		Zero-shot (16/8)		
	RE10K	ACID	RE10K-L	ACID-L	ScanNet	DL3DV-E	DL3DV-R
PSNR (dB) ↑							
c1 $L_{\max} = 4$	28.42	29.99	25.28	26.54	27.62	20.52	20.46
c2 $L_{\max} = 8$ (= b1)	28.59	30.08	**25.79**	**26.92**	28.60	21.29	21.20
c3 $L_{\max} = 12$	**28.67**	**30.14**	25.22	26.24	**28.76**	**21.51**	**21.46**
SSIM ↑							
c1 $L_{\max} = 4$	0.886	0.861	0.801	0.760	0.822	0.603	0.605
c2 $L_{\max} = 8$ (= b1)	0.890	0.863	**0.820**	**0.780**	0.843	0.636	0.636
c3 $L_{\max} = 12$	**0.892**	**0.865**	0.788	0.752	**0.847**	**0.648**	**0.649**
LPIPS ↓							
c1 $L_{\max} = 4$	0.113	0.134	0.194	0.215	0.214	0.322	0.317
c2 $L_{\max} = 8$ (= b1)	0.110	0.131	**0.180**	**0.201**	0.194	0.308	0.306
c3 $L_{\max} = 12$	**0.109**	**0.130**	0.209	0.228	**0.185**	**0.291**	**0.287**

Figure 8 extends the per-step view of App. A.1 to the remaining component ablations on the two zero-shot benchmarks. Without the gate prior the backbone render starts far lower and most of the gain arrives in the last steps, while the regularized gate front-loads quality so that intermediate depths remain usable out of distribution as well.

Table 7 is the halting sweep plotted in Fig. 5: zero-shot ScanNet, 800 images, $\tau \in \{0.02, \ldots, 0.98\}$ plus the full-depth point. Because the exit CDF is monotone in t, higher τ implies strictly later exit, giving the monotone depth-vs-τ relation in the mean-depth column. For visual results, check out the *Scaling* section on our website.

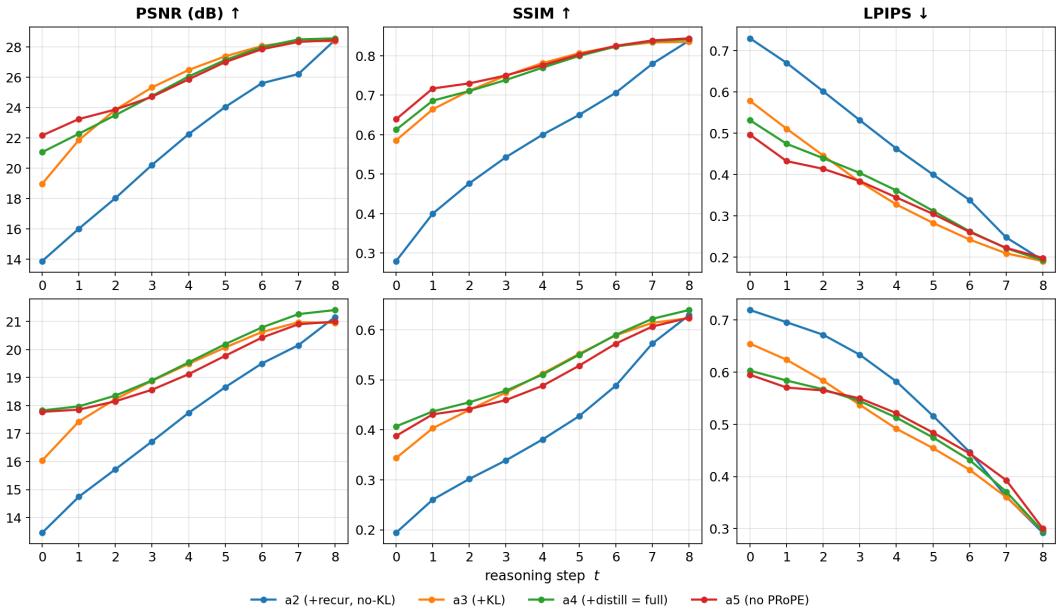


Figure 8: Per-step trajectories of the component ablations, zero-shot. *Top*: ScanNet. *Bottom*: DL3DV even split. All three metrics after each reasoning step t for a2 (recurrence, no KL), a3 (+KL), a4 (+distillation, full UVSM) and a5 (full, PRoPE off); each endpoint is the corresponding full-depth row of Table 2.

Table 7: **Halting-threshold sweep (ScanNet zero-shot, 800 images).** All three metrics improve monotonically with mean realized depth; adaptive halting spans mean depth 2.9–4.7 and full depth is the converged ceiling. Bold marks the full-depth headline operating point.

threshold τ	mean realized depth	PSNR ↑	SSIM ↑	LPIPS ↓
0.02	2.875	26.67	0.749	0.377
0.04	3.041	26.93	0.759	0.367
0.06	3.138	27.07	0.765	0.361
0.08	3.209	27.17	0.769	0.356
0.10	3.265	27.26	0.772	0.352
0.20	3.449	27.51	0.783	0.338
0.30	3.572	27.65	0.789	0.329
0.40	3.674	27.75	0.794	0.321
0.50	3.769	27.83	0.798	0.315
0.60	3.865	27.90	0.801	0.309
0.80	4.097	28.02	0.807	0.297
0.90	4.290	28.09	0.811	0.290
0.92	4.348	28.11	0.813	0.287
0.94	4.421	28.13	0.814	0.285
0.96	4.519	28.16	0.815	0.282
0.98	4.680	28.20	0.818	0.276
full depth	8.000	**28.56**	**0.841**	**0.193**

E QUALITATIVE RESULTS

Every sheet below is one scene under the matched protocol at full depth: rows are ground truth, UVSM, LVSM-dec-only, the encoder–decoder LVSM, SVSM, and iLRM; columns are the target views of that scene (8 for the zero-shot 16/8 protocol, 3 for the in-distribution 2/3 protocol). Baselines are retrained under the same data, effective batch size and iteration budget (Sec. 4). Figures 3 and 4 in the main text are single columns drawn from these sheets. Read together they show the same split the tables report: zero-shot, the baselines lose scene structure—doubled or transparent surfaces, dissolved thin geometry, and in iLRM's case near-total blur—while in distribution the methods are hard to tell apart except at fine texture.

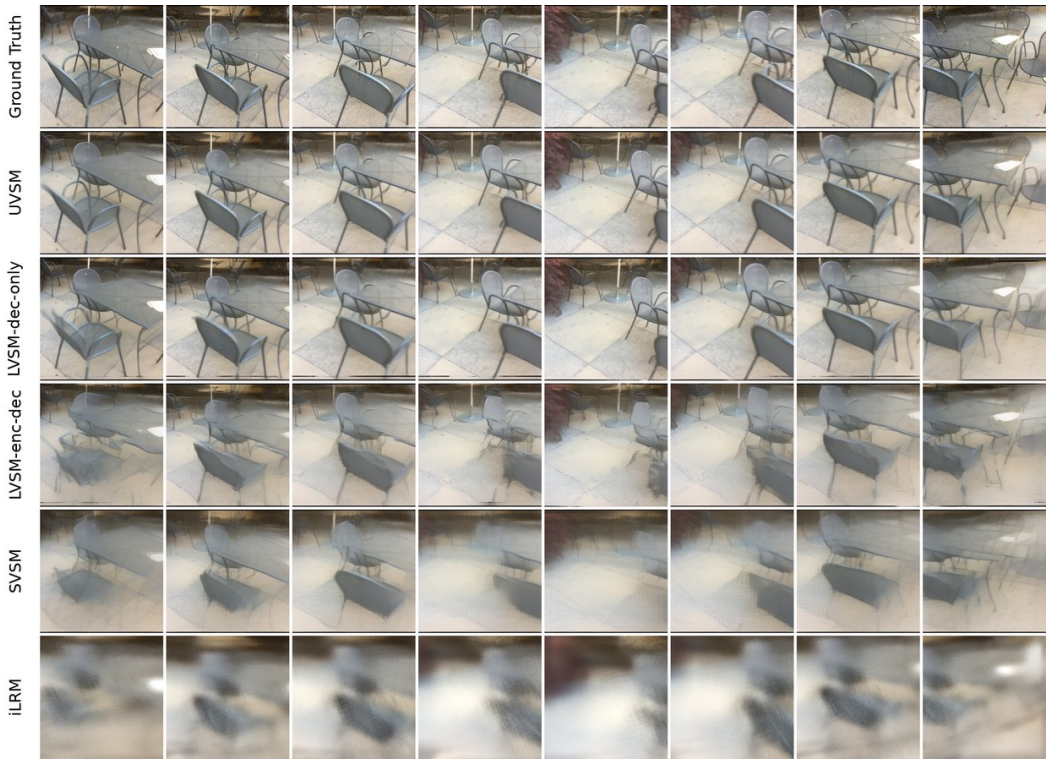


Figure 9: ScanNet, scene 000087 (zero-shot, 16 input / 8 target views).

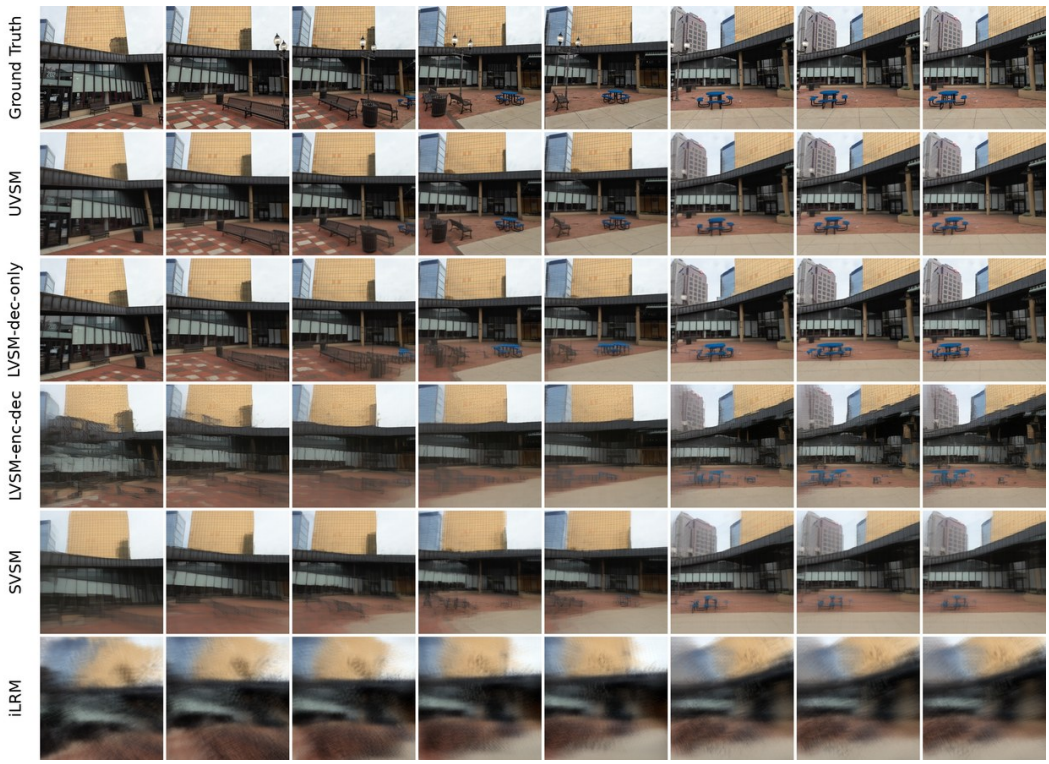


Figure 10: DL3DV even split, scene 000008 (zero-shot, 16 input / 8 target views).

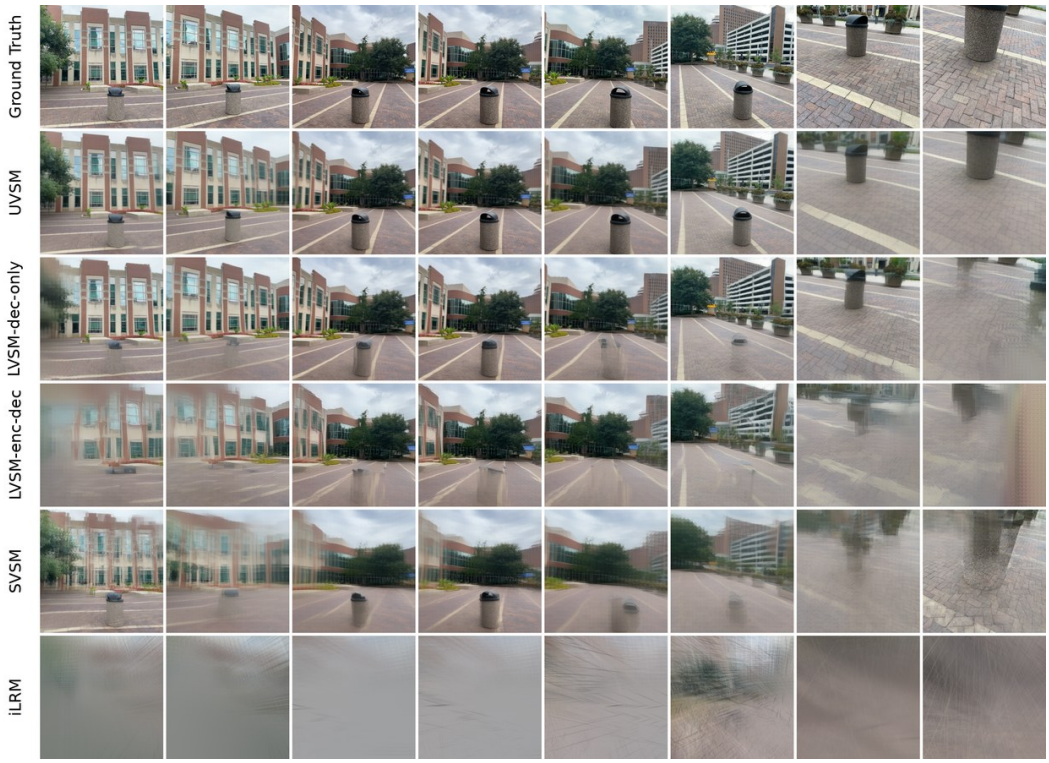


Figure 11: DL3DV even split, scene 000013 (zero-shot, 16 input / 8 target views).

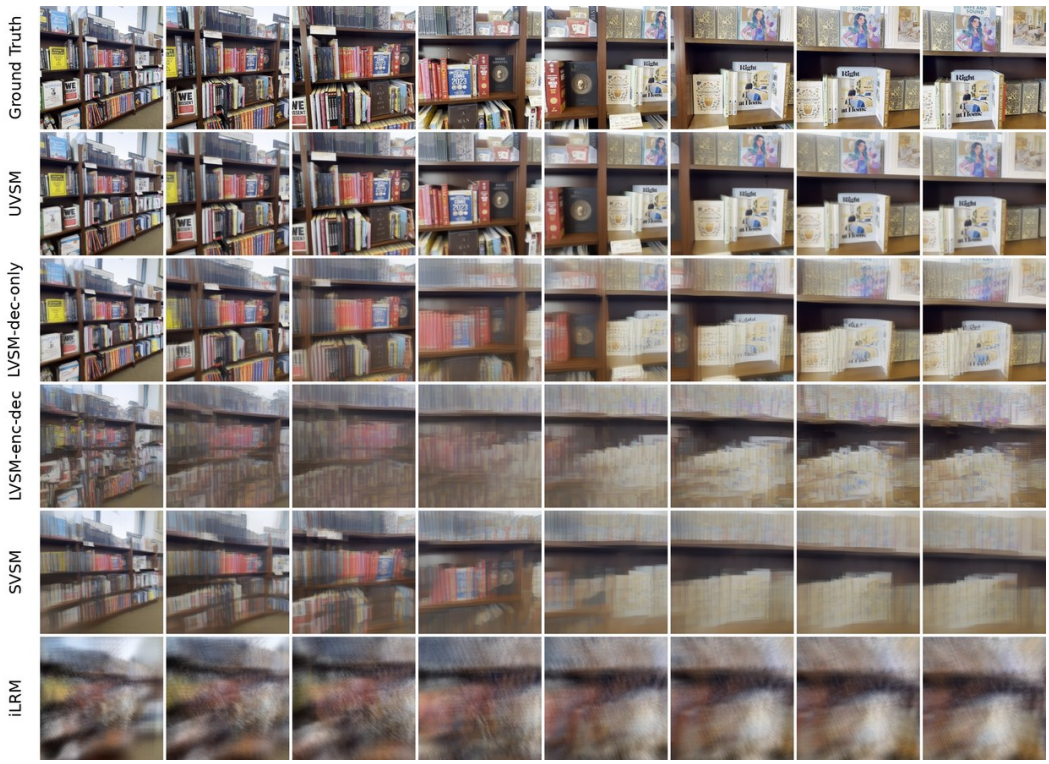


Figure 12: DL3DV even split, scene 000034 (zero-shot, 16 input / 8 target views).

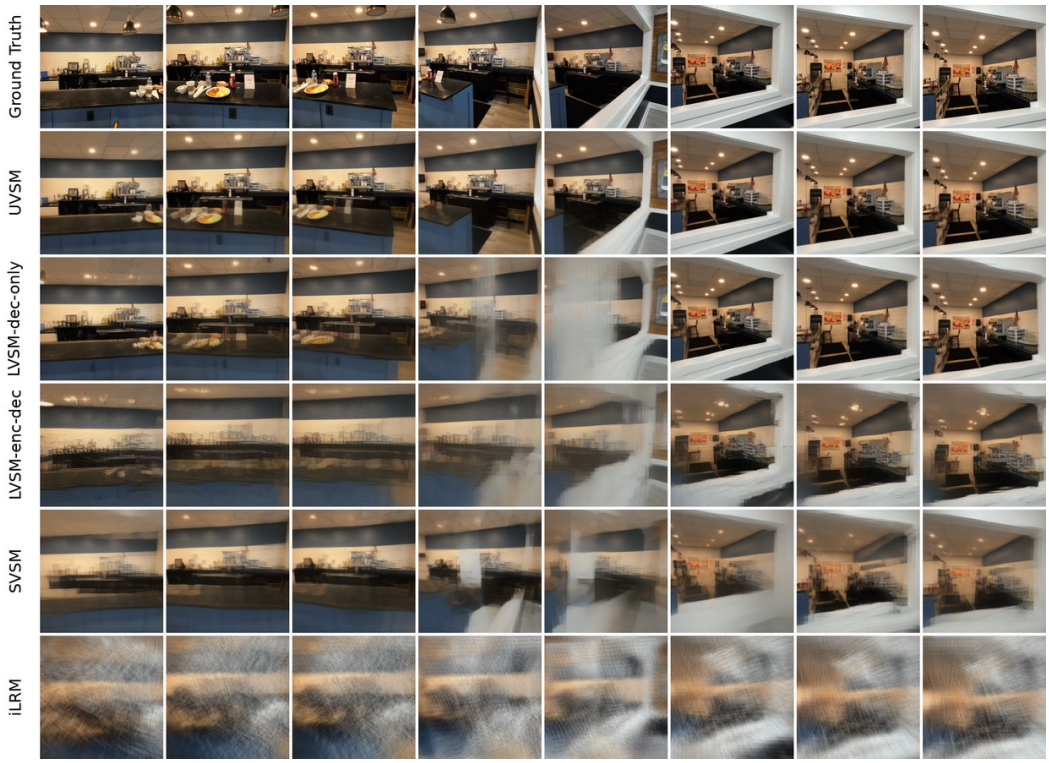


Figure 13: DL3DV even split, scene 000056 (zero-shot, 16 input / 8 target views).

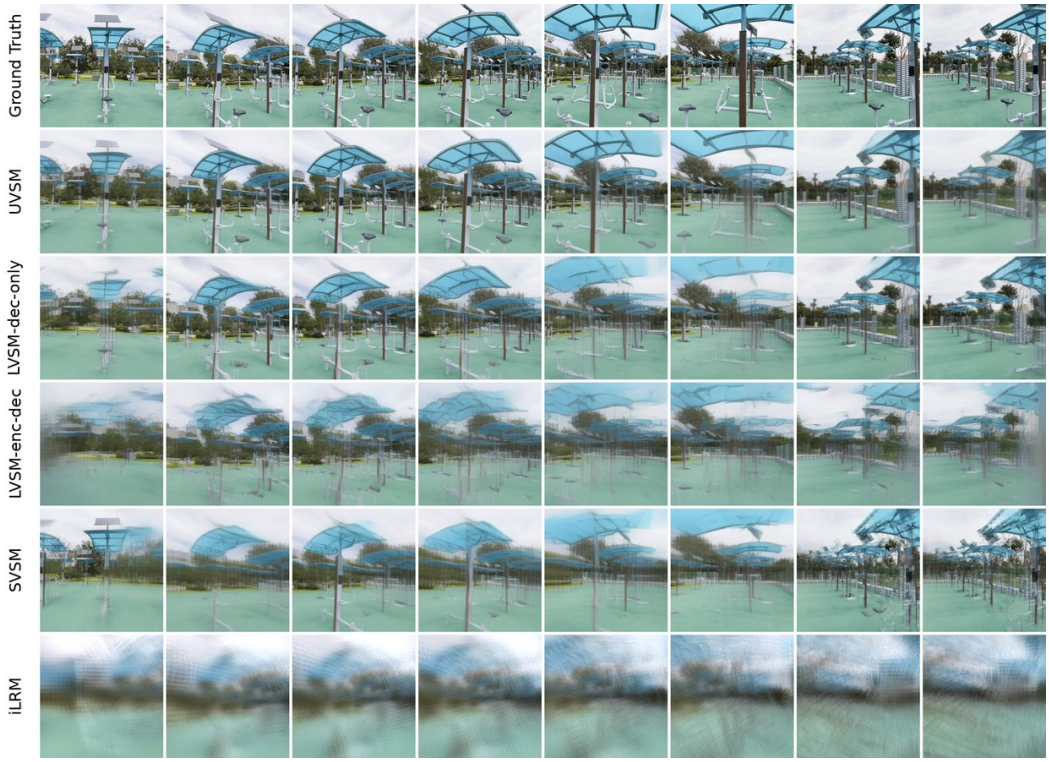


Figure 14: DL3DV even split, scene 000069 (zero-shot, 16 input / 8 target views).



Figure 15: DL3DV even split, scene 000072 (zero-shot, 16 input / 8 target views).

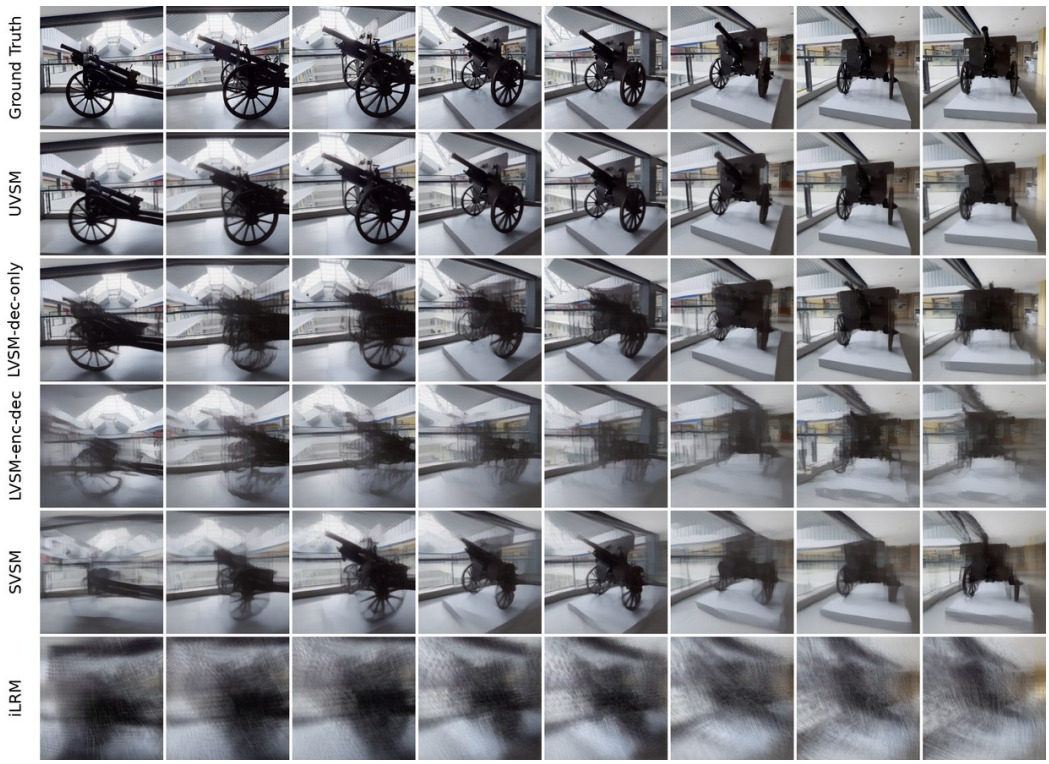


Figure 16: DL3DV random split, scene 000123 (zero-shot, 16 input / 8 target views).

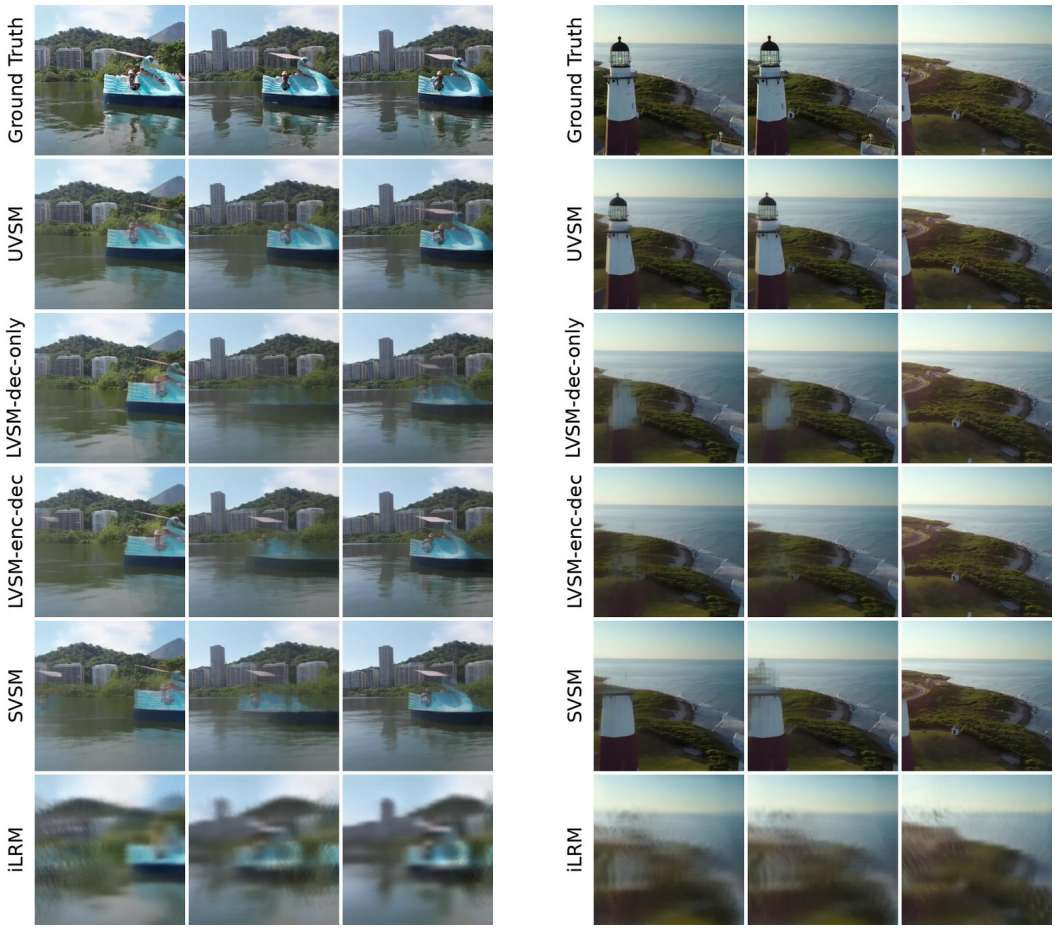


Figure 17: In-distribution sheets (2 input / 3 target views). ACID, scene 000705 ACID, scene 001034 (left to right).

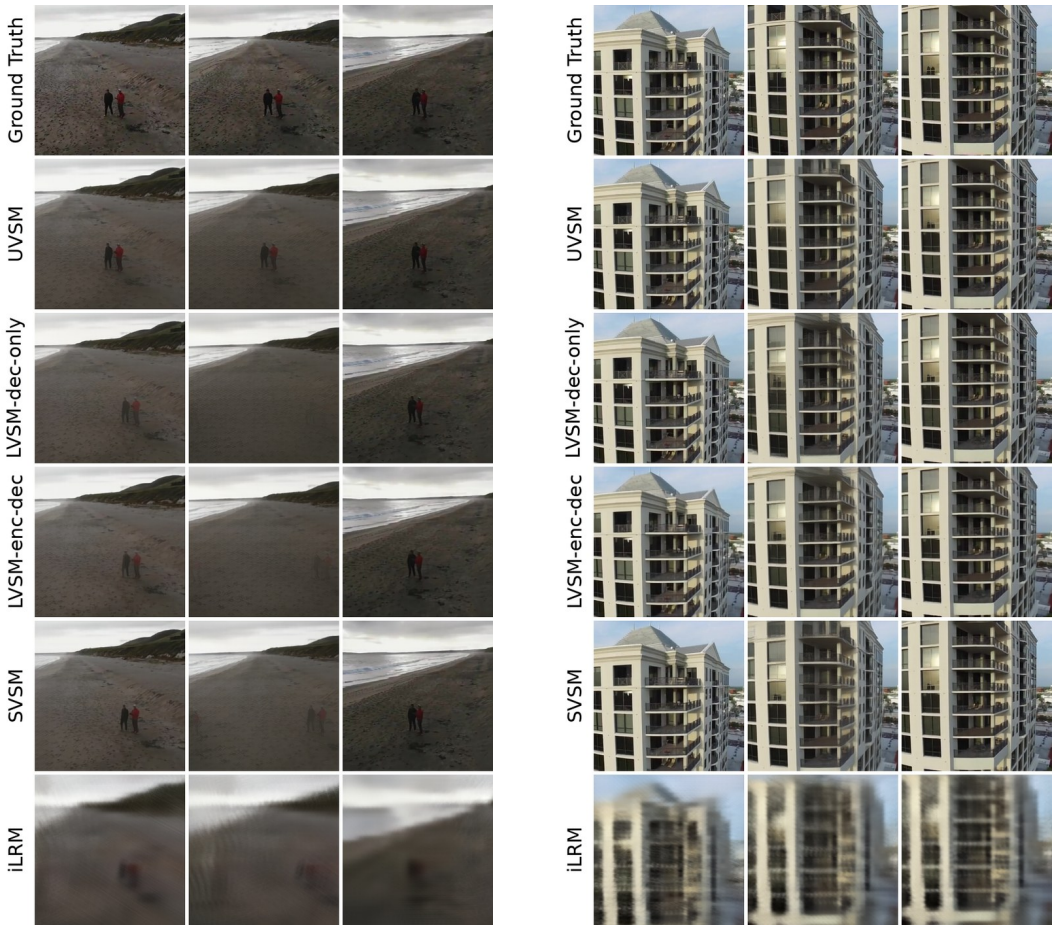


Figure 18: In-distribution sheets (2 input / 3 target views). ACID, scene 001078 RealEstate10K, scene 002567 (left to right).