FROM FEED-FORWARD TO FLOW: UNIFYING RECONSTRUCTION AND GENERATION IS EASIER THAN YOU THINK

Anonymous authors
Paper under double-blind review

ABSTRACT

Reconstruct where the images provide evidence, and generate where they do not: recent success of spatial world models such as Atlas (World Labs Team, 2026) highlights the value of unifying reconstruction and generation in one model. Yet the two have long lived in separate paradigms with distinctive failure modes: feed-forward reconstruction averages ambiguity into blur, while conditional generation invents plausible but scene-inconsistent detail.

In this work, we present a **unified flow-based formulation** for reconstruction and generation, where a shared clean-target predictor performs direct reconstruction at its single-step endpoint and unfolds conditional generation through multi-step flow. A controlled toy study reveals the mechanism: with a single step, the predictor collapses to the *conditional mean* just like feed-forward methods, favoring consistency over diversity. With multi-step inference, the fidelity of generated details grows with *context richness*: closer observations reduce ambiguity and yield better-matched details. We further instantiate the formulation in appearance and geometry 3D tasks. JiT-LVSM improves perceptual and distributional quality in novel view synthesis, while JUSt3R retains competitive single-step geometry prediction with additional multi-step inference capabilities that reduces veil and flying-pixel artifacts, producing cleaner surface structure with greater test-time compute. Together, they show that reconstruction and generation can share both a formulation and a backbone, with their behavior governed by denoising configuration—making unification surprisingly simple.

1 INTRODUCTION

Inferring a 3D scene from images requires a model both to recover what the observations support and to complete what they leave unknown. Modern systems typically address these two demands through different paradigms. Feed-forward reconstruction predicts a target view or scene geometry in a single pass (Charatan et al., 2024; Jin et al., 2025; Wang et al., 2024; 2025b), whereas conditional generation produces a target through iterative denoising (Watson et al., 2022; Chan et al., 2023; Kani et al., 2023). Their characteristic failures are markedly different: direct prediction can blur appearance, erase fine detail, or place geometry between competing surfaces, while generative inference produces sharp and plausible detail that may not faithfully correspond to the observed scene.

Bridging the boundary between these capabilities is becoming central to general-purpose world models. Atlas (World Labs Team, 2026), for example, brings generation and reconstruction into a shared spatial model: preserving what the observations show while imagining scene content beyond them. This reflects a broader goal for spatial intelligence—to reconstruct where the observations provide sufficient evidence and generate where information is missing. Reconstruction and generation should therefore not be treated only as separate tasks, but as complementary behaviors that a unified model must support.

This practical convergence raises a fundamental question: how can the goal of unifying reconstruction and generation be expressed in a common formulation that explains their different behaviors and capabilities? We address this question with a **unified flow-based formulation** that is surprisingly

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.

simple: direct reconstruction emerges at the one-step endpoint of a shared clean-target predictor, while conditional generation unfolds through multi-step flow.

Specifically, we make this connection precise by placing clean-target prediction along a noise-to-data flow (Lipman et al., 2022; Li & He, 2025). The predictor estimates a clean target X_0 from the observed context C and an intermediate flow state X_t, where the path starts from independent noise at $t = 0$ and ends at the target at $t = 1$. Reconstruction and generation thus predict the same target, and differ only in how many predictions are composed along the path. At the starting endpoint, X_t is pure noise and contains no target-specific information beyond the context. A one-step prediction from this endpoint therefore has the same Bayes-optimal action as a direct feed-forward predictor under the same supervised task loss. Under squared-error supervision, this common optimum is the conditional mean $\mathbb{E}[X_0 \mid C]$. The optimality of the mean is classical, but its consequence is easy to overlook: when multiple incompatible targets are consistent with the observations, this point estimate averages their competing details and need not resemble any individual solution. We refer to this mean-seeking effect as *conditional-mean bias*.

Multi-step flow moves beyond this context-only point estimate by conditioning each prediction on an intermediate flow state. For the ideal predictor, integrating the resulting state-dependent field transports noise toward the conditional target distribution, allowing an individual plausible solution to emerge. The consequences are concrete in both appearance and geometry. In novel view synthesis (NVS), averaging multiple compatible textures or boundaries can erase fine detail and produce blur, while such an averaged prediction may still receive a higher PSNR than a sharper generated view. In geometry prediction, uncertainty between foreground and background surfaces can place a point estimate between their depths, so that points float in the empty space between surfaces as veils or flying pixels, despite a small pointwise error. Multi-step flow inference can instead commit to a particular texture or surface, improving perceptual or structural quality even when conventional distortion metrics do not improve. Such commitment, however, is only as reliable as the available context: sparse observations may leave several solutions compatible with the inputs, while richer observations constrain which generated solution corresponds to the scene.

We study the formulation through a controlled mechanism experiment and two real 3D applications. First, a toy distribution with an analytic conditional mean verifies that feed-forward and one-step predictions recover the same endpoint solution, while multi-step generation from the same model produces individual conditional samples. Second, JiT-LVSM instantiates the formulation for novel view synthesis on LVSM (Jin et al., 2025), improving perceptual and distributional quality and showing how richer context brings generated detail into closer agreement with the reference scene. Third, JUSt3R carries the same formulation into pointmap prediction on DUSt3R (Wang et al., 2024): its one-step endpoint remains an effective direct predictor, while multi-step flow reduces veil and flying-pixel artifacts, producing cleaner surface structure even when pointwise depth error does not improve. Both models retain their feed-forward backbones with only a noisy target input, flow-time conditioning, and flow-matching training, allowing a single checkpoint to act as a direct predictor or generator, where inference steps provide a natural axis for test-time scaling. Together, these experiments connect the endpoint-to-flow formulation to complementary notions of quality across appearance and geometry.

Contributions. **1)** We present a unified flow-based formulation in which one common predictor supports feed-forward reconstruction and conditional generation through different flow configurations. **2)** We identify and explain the conditional-mean bias behind feed-forward prediction, prompting a rethink into the long-prevailing yet potentially misleading distortion scores. **3)** We instantiate the formulation in NVS and geometry with JiT-LVSM and JUSt3R, demonstrating simplicity, visual and structural benefits of generative inference.

2 Unifying Reconstruction and Generation through Flow

We revisit feed-forward prediction as the one-step endpoint of conditional flow. The same clean-target interface describes direct reconstruction and multi-step generation; their difference is how predictions are composed along the path. Formal statements, loss-dependent extensions, and proofs are given in Appendix B.

2.1 Conditional Clean-Target Prediction

Let C denote the observed context and $X_0 \sim p_{\mathrm{data}}(\cdot \mid C)$ the clean target, such as a novel-view image or a pointmap. Following flow matching (Lipman et al., 2022), we use the noise-to-data path

$$X_t = (1-t)\epsilon + tX_0, \qquad \epsilon \sim p_0, \qquad \epsilon \perp X_0 \mid C, \tag{1}$$

where the base noise is drawn independently of (X_0, C). Time runs from noise at $t = 0$ to data at $t = 1$; X_0 always denotes the clean target. The model predicts that target directly (Li & He, 2025), inducing a flow velocity:

$$\widehat{X}_0 = f_\theta(X_t, t, C), \qquad v_\theta(x, t, C) = \frac{f_\theta(x, t, C) - x}{1-t}, \quad t < 1. \tag{2}$$

Flow matching supervises this velocity against $X_0 - \epsilon$, equivalently a time-weighted squared error on the clean prediction. Both reconstruction and generation use this prediction interface.

2.2 From the Feed-Forward Endpoint to Multi-Step Flow

Single-step collapse. A feed-forward predictor receives only context; at $t = 0$, a denoiser additionally receives independent noise. Under the same squared-error loss, both therefore have the same population optimum:

$$f^\star(\epsilon, 0, C) = \mathbb{E}[X_0 \mid \epsilon, C] = \mathbb{E}[X_0 \mid C]. \tag{3}$$

One Euler step from noise to the data endpoint returns this prediction directly. This is the *conditional-mean bias*: when several targets fit the context, a one-step estimate averages their competing explanations. Other task losses select different summaries, but independent noise still leaves the optimal-action set unchanged (Proposition 1).

Multi-step flow. At intermediate times, the training state contains target information, so the optimal squared-error predictor $\mathbb{E}[X_0 \mid X_t, C]$ depends on the current state. The corresponding ideal velocity transports noise toward the conditional target distribution. Thus a sequence of conditional means need not produce the context-only mean: a single jump returns a summary, while integrating the state-dependent field can generate a sample. At inference, intermediate states propagate the model's evolving hypothesis; they do not supply new observations. Denoising depth controls how many prediction steps compose that hypothesis. A model trained along the flow can therefore span reconstruction-like and generation-like behavior through its inference configuration.

2.3 Implications for 3D Prediction

When context leaves several plausible explanations, averaging can blur image detail or compromise surface structure. It can nevertheless improve pointwise distortion, so PSNR alone may reward the very behavior that makes a prediction look worse. Our toy experiment makes this mechanism visible, and the NVS and geometry experiments examine its perceptual and structural consequences.

Context and depth play complementary roles. Richer observations constrain which scene explanations are plausible; additional flow steps express the ambiguity that remains. Under sparse context, a sharp sample may still choose an incorrect completion. With richer context, generated detail can agree more closely with the observed scene. This shared relationship applies to both appearance and geometry, and gives multi-pass denoising a practical role as test-time computation.

3 Unifying Feed-Forward Reconstruction and Generation

We test the unified flow formulation in three settings that span a range of posterior widths. A toy distribution with an exact posterior lets us vary its width directly and compare against analytic references (3.1). Novel view synthesis has a broad posterior whose width varies with source-view overlap. We convert LVSM into JiT-LVSM and compare the two under sparse and rich context (3.2). Pointmap prediction has a narrow posterior that is ambiguous mainly at depth discontinuities. We convert DUSt3R into JUSt3R and evaluate one checkpoint with one and multiple steps (3.3). In both 3D tasks we read distortion metrics together with perceptual or structural ones, then demonstrate how a common prediction interface supports reconstruction and generation across different output spaces.

3.1 TOY EXPERIMENT

Setup. We first test the formulation where every quantity of interest is known exactly. Each sample has a hidden mode $z \in \{0, \dots, K-1\}$ with $K = 8$. The target $X_0 \in \mathbb{R}^2$ is sampled from an isotropic Gaussian ($\sigma_m = 0.15$) centered at μ_z, where the mode centers μ_k lie equally spaced on a circle of radius 2 (marked $\times$ in Figure 1). Colors throughout the figure indicate the ground-truth mode z. A context signal c noisily identifies the mode, and its reliability $r \in [0, 1]$ directly sets the posterior width: at $r = 0$ all modes are nearly equally plausible, and at $r = 1$ the context identifies the correct one. Because the mixture is known, both the conditional mean $\mathbb{E}[X_0 \mid c]$ (`mean`) and exact posterior samples (`exact`) are available in closed form, providing references for Proposition 1 that are unavailable in the real 3D tasks.

For each r, we train a context-only regressor $f(c) \to X_0$ (`feed-forward`) and a single clean-target flow model $f_\theta(X_t, t, c)$ with the same flow-matching objective used in the 3D tasks. We evaluate the *same* flow checkpoint with one Euler step (`flow-1`), which returns $f_\theta(\epsilon, 0, c)$, and with 15 Euler steps (`flow-15`). In Figure 1, the left block of each row samples all modes jointly, while the right block fixes a single ground-truth mode for every sample, so that wrong-mode samples are directly visible as points near an incorrect center.

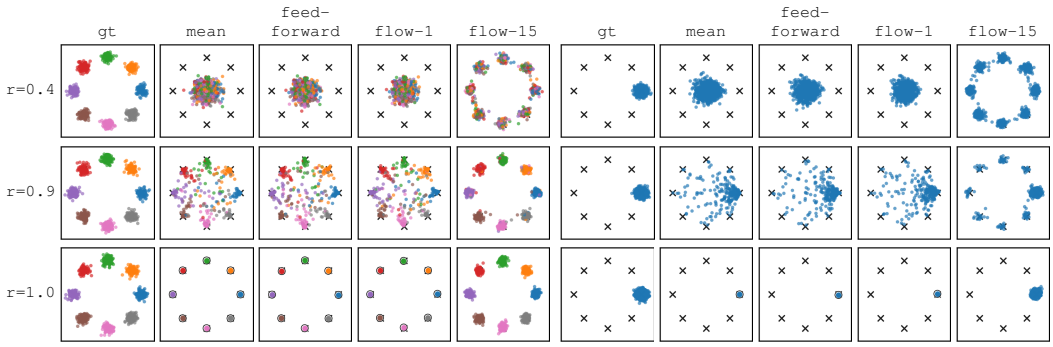

Figure 1: Controlled toy experiment. Rows increase context reliability r. Left blocks show all modes, while right blocks fix one hidden mode for every sample, so wrong-mode samples appear near an incorrect center. Columns: ground truth, the analytic conditional mean, feed-forward prediction, and one flow model evaluated with one and 15 Euler steps. One step tracks the mean, which can lie between modes; 15 steps from the same model sample the posterior.

One step recovers the conditional mean. In every row of Figure 1, the `flow-1` column is visually indistinguishable from the `mean` column, as is `feed-forward`. The flow model receives an extra noise input, yet its one-step output collapses onto the same point estimate, as Proposition 1 predicts.

More steps commit to the data manifold. At $r = 0.4$, the `mean`, `feed-forward`, and `flow-1` columns all place most of their predictions in the interior of the circle, between the modes and far from any ground-truth cluster. The best squared-error summary thus need not resemble the data it predicts. Run with 15 steps, the same checkpoint (`flow-15`) instead places its samples back on the circle, in tight clusters that resemble the `gt` column. This is the mechanism behind the metric disagreements in the 3D experiments below: pointwise distortion rewards the averaged prediction, while perceptual and structural measures reveal what it loses.

Posterior width decides whether the commitment is correct. The fixed-mode blocks show which modes the samples come from. At $r = 0.4$, `flow-15` spreads its samples across many modes even though every sample shares one hidden mode: the samples are realistic but often wrong, because the context cannot tell the modes apart. At $r = 0.9$ most samples fall on the correct mode, and at $r = 1.0$ all of them do, while the one-step columns converge to the correct center. As context narrows the posterior, multi-step samples become both realistic and correct.

Quantitative results and experiment details are in Appendix C, further confirming our observations.

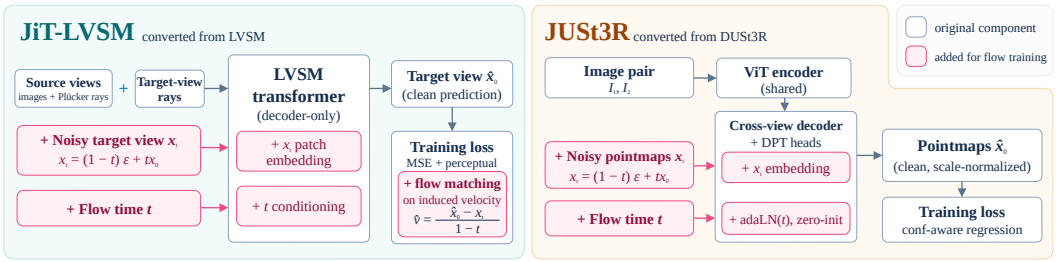

Figure 2: From feed-forward to flow. Both models keep the original backbone and add only a noisy target x_t and flow time t; one checkpoint is both a one-step predictor and a multi-step generator.

3.2 Novel View Synthesis with JiT-LVSM

Setup. We instantiate the formulation with the decoder-only LVSM backbone (Jin et al., 2025) on RealEstate10K (Zhou et al., 2018) and DL3DV (Ling et al., 2024). JiT-LVSM keeps LVSM's source-image and camera-ray interface and adds a noisy target image and flow-time conditioning (Figure 2, left); it predicts the clean image and is trained through the induced velocity with flow matching and image-space supervision (Li & He, 2025). At inference, it generates the target view by integrating the learned flow from noise with Heun's method in 10 steps. Both models share the same backbone, resolution, and conditioning inputs, so the comparison isolates the effect of the flow formulation. We also report PixelSplat (Charatan et al., 2024) and NoPoSplat (Ye et al., 2024a) as feed-forward references with explicit 3D Gaussian representations. Architecture and training details, together with the target-side noise sanity check, are given in Appendix D.

Posterior width depends on context. When the source views leave the target underconstrained, the conditional target distribution is broad, and it is broad over large regions of the image rather than only at boundaries. In this regime, LVSM produces conservative, blurred predictions in ambiguous regions, as expected from the averaging mechanism of Section 2.2. Multi-step flow inference lets JiT-LVSM commit to sharp detail instead, but under sparse context the committed detail may belong to a plausible completion that the observed scene does not support. Richer context narrows the posterior, so the same inference produces detail that agrees more closely with the reference. We therefore compare the two models under sparse and rich context.

Qualitative comparison. Figures 3 and 4 show representative scenes. Across examples, JiT-LVSM produces sharper and more structured target views, while LVSM blurs extrapolated or ambiguous content. Under sparse context, this blur is a conservative choice. The prediction looks less realistic, but it avoids committing to unsupported high-frequency detail. JiT-LVSM is more realistic in the same setting, yet some of its added detail does not match the reference view. Under rich context, the source views constrain the target more tightly, and JiT-LVSM's detail more often agrees with the reference. Appendix F shows 60 randomly selected DL3DV scenes and 60 RealEstate10K scenes.

Quantitative comparison. Table 1 compares JiT-LVSM with LVSM (Jin et al., 2025), PixelSplat (Charatan et al., 2024), and NoPoSplat (Ye et al., 2024a) under sparse and rich context on both datasets. PSNR and SSIM measure pointwise agreement with the reference, which an averaged

Method	RealEstate10K (Zhou et al., 2018)								DL3DV (Ling et al., 2024)							
	Sparse Context				Rich Context				Sparse Context				Rich Context			
	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
PixelSplat	17.18	0.566	0.496	101.2	25.51	0.867	0.126	3.102	16.78	**0.512**	0.511	127.1	21.02	0.725	0.220	5.276
NoPoSplat	16.02	0.511	0.588	98.23	25.46	0.854	0.137	3.017	15.17	0.497	0.602	172.3	21.50	0.751	0.237	5.145
LVSM	**19.78**	**0.601**	0.331	32.84	27.14	**0.875**	0.118	2.979	**16.81**	0.410	0.482	108.7	22.29	0.727	0.201	3.996
JiT-LVSM	17.94	0.593	**0.275**	**21.45**	**27.77**	0.872	**0.117**	**2.977**	15.13	0.446	**0.421**	**52.02**	**22.93**	**0.769**	**0.179**	**3.292**

Table 1: Novel view synthesis on RealEstate10K (Zhou et al., 2018) and DL3DV (Ling et al., 2024). PSNR and SSIM measure agreement with the reference view; LPIPS and FID measure perceptual and distributional quality. **Bold**: best; underline: second best.

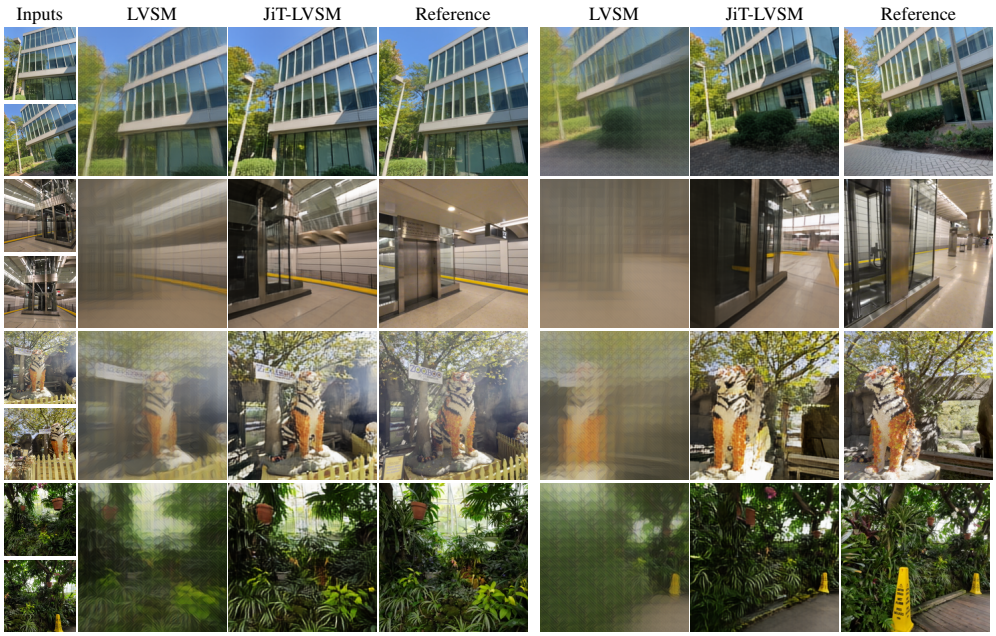

Figure 3: Sparse-context comparison on DL3DV (Ling et al., 2024). LVSM blurs regions the inputs do not constrain; JiT-LVSM commits to sharp detail. This visual improvement need not raise PSNR, as the aggregate sparse-context results in Table 1 also show.

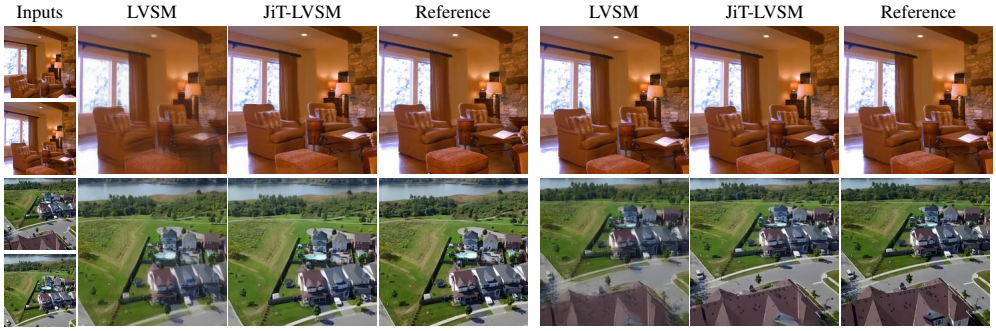

Figure 4: Rich-context comparison on RealEstate10K (Zhou et al., 2018). With more constraining observations, JiT-LVSM's detail agrees more closely with the reference.

prediction can score well on. LPIPS and FID measure whether the generated detail looks realistic. *Sparse context.* JiT-LVSM gives the best LPIPS and FID on both datasets, but its PSNR is 1.7–1.8 dB lower than LVSM. This is the trade-off predicted by Section 2.2: LVSM approximates the PSNR-optimal conditional mean, and a sample from a broad posterior has higher squared error than that mean even when it looks more realistic. *Rich context.* The trade-off largely disappears: JiT-LVSM improves PSNR over LVSM by about 0.6 dB on both datasets and remains best in LPIPS and FID, by a clear margin on DL3DV and a small one on RealEstate10K. As the posterior narrows, flow inference keeps its realism while its committed detail increasingly agrees with the particular target scene.

Measuring context richness. To make "sparse" and "rich" measurable, we estimate how much the source views overlap using dense correspondences from RoMA (Edstedt et al., 2024), as a proxy for how strongly the inputs constrain the target. The probe describes the available evidence instead of estimating the posterior entropy directly. Table 2 shows that source-view overlap is much weaker in our DL3DV evaluation set than in RealEstate10K: the median

Metric	DL3DV	RealEstate10K
# scenes	140	6424
RoMA mean	0.111	0.249
RoMA median	0.056	0.237
Conf. > 0.5 mean	0.049	0.142
Grid cov. > 0.5 med.	0.188	0.688

Table 2: Source-view overlap measured with RoMA on the two evaluation sets.

fraction of grid cells with confident matches is 0.188 versus 0.688. Weaker overlap indicates a broader posterior, and it is on DL3DV that flow inference changes the output distribution most. FID is roughly halved under sparse context, against a one-third reduction on RealEstate10K. The next experiment turns to geometry, where the posterior is narrow except at boundaries and the effect of committing to one solution can be measured directly.

3.3 Geometry Prediction with JUSt3R

Setup. We instantiate the same formulation in pointmap space with DUSt3R (Wang et al., 2024). Our **JUSt3R** keeps DUSt3R's backbone, training data, and schedule, and adds a noisy target pointmap x_t through a light decoder-token embedding and flow time t through adaLN (Figure 2, right). It predicts the clean pointmap and is trained with DUSt3R's own confidence-weighted regression loss. We denote inference with k steps from a single checkpoint by $\mathrm{JUSt3R}_k$: $k = 1$ is the direct-prediction endpoint, and larger k composes predictions along the flow, reusing one context encoding. We compare against DUSt3R retrained by ourselves, which is the controlled baseline, and report the released checkpoint (DUSt3R*) for reference. Implementation and evaluation details are in Appendix E.

Where the geometric posterior is ambiguous. Unlike novel view synthesis, pointmap prediction has a posterior that is narrow over most of the image because visible surfaces are well constrained by the input and by learned geometric priors. Ambiguity concentrates at depth discontinuities, mixed pixels, and thin structures, where a pixel may belong to either the foreground or the background surface. DUSt3R's loss is a confidence-weighted, scale-normalized $L_{2,1}$ regression, so it still yields a single point estimate per pixel. Near a boundary, where the pixel may belong to either surface and the edge position itself is uncertain, this estimate is often placed between the two surfaces, producing the veils and flying pixels typical of feed-forward pointmap regression. We therefore expect multi-step inference to change pointwise accuracy little but to act locally at boundaries.

Evaluating surface structure. AbsRel and $\delta_{1.25}$ compare each depth against a reference (Wang et al., 2025c; Zhang et al., 2025) and are dominated by the well-constrained interior, so they can hide exactly these boundary errors. We therefore also report the *veil rate*, the fraction of ground-truth depth discontinuities where the prediction falls in the middle of the gap between the two surfaces. Meanwhile, a ground-truth-free *flying-pixel rate* that flags points on locally grazing geometry (Pomerleau et al., 2013) and *BF1* for depth-edge localization (Bochkovskiy et al., 2025; Xu et al., 2026) are included. Veil and flying-pixel rates reward committing to a surface but not choosing the correct one, so we read them together with BF1 and AbsRel.

Qualitative comparison. Figure 5 compares DUSt3R with one-, ten-, and fifty-step JUSt3R on two NYU-v2 scenes and one BlendedMVS scene. DUSt3R and $\mathrm{JUSt3R}_1$ produce streaks and stretched surfaces around furniture, wall openings, and the carved column, whereas multi-step predictions separate these surfaces cleanly while preserving the overall layout. Figure 6 shows fifty-step outputs across object, indoor, and outdoor inputs, and Appendix G provides further comparisons.

Benchmark	AbsRel ↓				$\delta_{1.25}$ ↑				BF1 ↑ (×100)				veil ↓				flying ↓			
	DUSt3R*	DUSt3R	JUSt3R$_1$	JUSt3R$_{10}$	DUSt3R*	DUSt3R	JUSt3R$_1$	JUSt3R$_{10}$	DUSt3R*	DUSt3R	JUSt3R$_1$	JUSt3R$_{10}$	DUSt3R*	DUSt3R	JUSt3R$_1$	JUSt3R$_{10}$	DUSt3R*	DUSt3R	JUSt3R$_1$	JUSt3R$_{10}$
NYU-v2	.080	.075	**.070**	.071	.907	.915	**.925**	**.925**	5.17	7.10	8.03	**8.84**	.220	.215	.219	**.186**	.041	.047	.037	**.023**
TUM	.180	.177	**.165**	.178	.774	.750	**.782**	.749	2.29	2.09	3.34	**4.16**	.271	.273	.231	**.169**	.027	.040	.038	**.022**
Bonn	.141	**.095**	.096	.105	.825	**.904**	.902	.885	1.88	2.71	**2.83**	2.57	.275	.230	.220	**.144**	.016	.024	.021	**.014**
Sintel	.413	.418	**.400**	.402	.560	.533	**.560**	.558	.36	.16	.36	**1.24**	.191	.210	.188	**.185**	.023	.030	.030	**.017**
KITTI	.112	.098	**.096**	.109	.863	.897	**.899**	.873	*sparse GT*				*sparse GT*				.109	.107	.106	**.089**

Table 3: Zero-shot depth on NYU-v2 (Silberman et al., 2012), TUM (Sturm et al., 2012), Bonn (Palazzolo et al., 2019), Sintel (Butler et al., 2012), and KITTI (Geiger et al., 2013). DUSt3R is retrained with our pipeline and is the controlled baseline; $\mathrm{JUSt3R}_1$ and $\mathrm{JUSt3R}_{10}$ are the same checkpoint with one and ten steps. **Bold**: best; underline: second best, ranked among these three. The released DUSt3R 512-DPT checkpoint (DUSt3R*, gray) is shown for reference. Veil rate and BF1 require dense ground truth and are omitted on KITTI.



Figure 5: Point-cloud comparison on two NYU-v2 scenes and one BlendedMVS scene (top to bottom), rendered from a 14° offset view. DUSt3R and one-step JUSt3R exhibit streaks and stretched surfaces near depth boundaries, where the conditional mean falls between surfaces; multi-step inference from the same JUSt3R checkpoint separates them.



Figure 6: Pointmaps from JUSt3R$_{50}$ on object, indoor, and outdoor inputs.

Quantitative comparison. Table 3 reports five zero-shot depth benchmarks. *The one-step endpoint is a competitive direct predictor.* With a single decoder pass, JUSt3R$_1$ improves AbsRel over the retrained DUSt3R on four of the five datasets and improves $\delta_{1.25}$ on four. Therefore training along the flow does not cost direct-prediction accuracy. *More steps act at boundaries.* Moving from JUSt3R$_1$ to JUSt3R$_{10}$ with the same checkpoint lowers the veil rate on all four datasets with dense ground truth, lowers the flying-pixel rate on all five (by 16–43%), and raises BF1 on three of four. Pointwise accuracy moves in the opposite direction but only slightly. This asymmetry is what a narrow posterior with localized ambiguity predicts. The interior, which dominates AbsRel, was already close to any sample, while the boundaries, where the mean falls between surfaces, are where committing to one surface matters. It mirrors the image-space perception–distortion trade-off (Blau & Michaeli, 2018), and contrasts with sparse-context NVS, where a broad posterior makes the same trade-off global.

3.4 Empirical Takeaways

Read together, the three settings support the formulation in Section 2 from complementary angles. The toy experiment checks them against exact references, and the two 3D tasks show how they surface in real output spaces with very different posteriors.

Reconstruction is the one-step endpoint of the flow. In the toy, one Euler step reproduces the analytic conditional mean as closely as a dedicated feed-forward regressor, despite the added noise. JUSt3R$_1$ does the same on a real backbone, improving AbsRel over the retrained DUSt3R on four of five benchmarks. Conversion to flow thus does not trade away reconstruction.

Committing to a solution trades distortion for realism. With more steps, the models commit to single solutions: toy samples return to the data manifold, JiT-LVSM halves FID on sparse-context DL3DV, and JUSt3R removes veils and flying pixels. In each case, pointwise distortion rises (up to 1.8 dB PSNR in NVS, 0.013 AbsRel in geometry). The mean fails alike across output spaces, averaging incompatible explanations into points inside the circle, blur in images, and points floating between surfaces. Hence PSNR or AbsRel favor the conditional mean and understate what generation contributes, while perceptual and structural measures hide what it costs; a fair comparison needs both.

Posterior width sets the price of commitment. The trade-off grows with posterior width, which has two sources. The first is context: as reliability rises in the toy, samples become both realistic and correct, and the PSNR cost of JiT-LVSM under sparse context turns into a gain under rich context. The second is the task: pointmaps are ambiguous mainly at depth discontinuities, so multi-step inference acts locally, whereas sparse-context NVS is ambiguous over large regions. Width also sets the risk of generation: under a broad posterior, a committed sample can be realistic but wrong, as in the toy's fixed-mode panels and sparse-context NVS. Generation does not substitute for missing evidence; it chooses among the completions the evidence leaves open.

Denoising depth is a test-time choice. Because one checkpoint serves both roles, the number of steps can be chosen at inference without retraining. A single step gives the lower-distortion estimate that downstream measurement or alignment may prefer; more steps give sharper, more committed outputs at additional decoder cost, which suits applications that value realism or clean structure. Which choice is right depends on how strongly the observations constrain the scene, and since that constraint varies across inputs and even across regions of one output, the choice need not be global.

4 Conclusion

From feed-forward to flow, the gap between reconstruction and generation is smaller than it appears. We present a unified flow-based formulation in which direct reconstruction occupies the single-step noise endpoint and conditional generation composes state-dependent clean-target predictions along the flow. This relationship explains why feed-forward methods can fail despite strong distortion scores: under squared error, their conditional-mean optimum may average incompatible explanations into an output that resembles none of them. A controlled toy experiment exposes this bias, while JiT-LVSM and JUSt3R connect the same perspective to sharper novel views and cleaner geometric structure. Together, these results make unification concrete across appearance and geometry, using familiar reconstruction backbones and different denoising configurations. They also reveal a useful consequence: multi-pass flow inference supplies additional test-time computation, allowing a fixed model to express detail that one-step prediction can average away. Context constrains that detail; distortion alone does not tell us whether it is useful.

Limitations and future directions. Our experiments cover NVS and geometry prediction at a fixed set of denoising depths. Larger feed-forward architectures such as VGGT (Wang et al., 2025b), and the choice of depth per input or spatial region, offer natural extensions of the formulation. The analysis concerns direct task-loss prediction and flow matching; its one-step endpoint result does not preclude one-step generators trained with different objectives. Recent consistency models (Song et al., 2023), mean-flow models (Geng et al., 2025; Lu et al., 2026), and drifting models (Deng et al., 2026) motivate a further direction: retaining the benefits of generative inference while reducing the computation needed to obtain them.

AI USE STATEMENT

Generative AI tools were used to improve the readability of the manuscript and to assist with LaTeX formatting. All AI-assisted content was reviewed and revised by the authors.

REPRODUCIBILITY STATEMENT

Dataset construction, preprocessing, hyperparameters, training, evaluation and additional implementation details are documented in the main paper and appendix. Code, dataset and pretrained checkpoints will be released upon publication.

ETHICS STATEMENT

The work is a methodological and empirical study of reconstruction and generation behavior. We are not aware of any direct ethical concerns arising from this work.

REFERENCES

Mohammad Asim, Christopher Wewer, and Jan Eric Lenssen. Scenetok: A compressed, diffusable token space for 3d scenes. *arXiv preprint arXiv:2602.18882*, 2026.

James O Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer Science & Business Media, 1985.

Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6228–6237, 2018.

Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *International Conference on Machine Learning*, pp. 675–685. PMLR, 2019.

Alexey Bochkovskiy, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *International Conference on Learning Representations*, volume 2025, pp. 75602–75637, 2025.

Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pp. 611–625. Springer, 2012.

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4217–4229, 2023.

Jiahao Chang, Chongjie Ye, Yushuang Wu, Yuantao Chen, Yidan Zhang, Zhongjin Luo, Chenghong Li, Yihao Zhi, and Xiaoguang Han. Reconviagen: Towards accurate multi-view 3d object reconstruction via generation. *arXiv preprint arXiv:2510.23306*, 2025.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19457–19467, 2024.

Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 14124–14133, 2021.

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. In *European conference on computer vision*, pp. 370–386. Springer, 2024.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19790–19800, 2024.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496): 1602–1614, 2011.

Dror Freirich, Tomer Michaeli, and Ron Meir. A theory of the distortion-perception tradeoff in wasserstein space. *Advances in Neural Information Processing Systems*, 34:25661–25672, 2021.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, et al. The elements of statistical learning, 2009.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, et al. Rayzer: A self-supervised large view synthesis model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4918–4929, 2025.

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. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=QQBPWtvtcn.

Bharath Raj Nagoor Kani, Hsin-Ying Lee, Sergey Tulyakov, and Shubham Tulsiani. Upfusion: Novel view diffusion from unposed sparse view observations. *arXiv preprint arXiv:2312.06661*, 2023.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in neural information processing systems*, 35:23593–23606, 2022.

Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9492–9502, 2024.

Thanh-Tung Le, Tuan Pham, Tung Nguyen, Deying Kong, Xiaohui Xie, and Stephan Mandt. Umami: Unifying masked autoregressive models and deterministic rendering for view synthesis. *arXiv preprint arXiv:2512.20107*, 2025.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pp. 71–91. Springer, 2024.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9298–9309, 2023.

Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. *arXiv preprint arXiv:2601.22158*, 2026.

Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, and Cyrill Stachniss. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7855–7862. IEEE, 2019.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

François Pomerleau, Francis Colas, Roland Siegwart, and Stéphane Magnenat. Comparing icp variants on real-world data sets: Open-source library and experimental protocol. *Autonomous robots*, 34(3):133–148, 2013.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pp. 746–760. Springer, 2012.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pp. 32211–32252. PMLR, 2023.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 573–580. IEEE, 2012.

Haoru Wang, Kai Ye, Yangyan Li, Wenzheng Chen, and Baoquan Chen. The less you depend, the more you learn: Synthesizing novel views from sparse, unposed images without any 3d knowledge. *arXiv preprint arXiv:2506.09885*, 2025a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025b.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2021.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10510–10522, 2025c.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20697–20709, 2024.

Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022.

World Labs Team. Atlas: A world model for spatial intelligence. *World Labs Blog*, September 2026. URL `https://www.worldlabs.ai/blog/atlas`. September 1, 2026.

Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21551–21561, 2024.

Mochu Xiang, Zhelun Shen, Xuesong Li, Jiahui Ren, Jing Zhang, Chen Zhao, Shanshan Liu, Haocheng Feng, Jingdong Wang, and Yuchao Dai. Rng: A unified transformer for complete 3d modeling from partial observations. *arXiv preprint arXiv:2603.01194*, 2026.

Haofei Xu, Rundi Wu, Philipp Henzler, Nikolai Kalischek, Michael Oechsle, Fabian Manhardt, Marc Pollefeys, Andreas Geiger, Federico Tombari, and Michael Niemeyer. Pointdit: Pixel-space diffusion for monocular geometry estimation. *arXiv preprint arXiv:2607.02515*, 2026.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1790–1799, 2020.

Botao Ye, Sifei Liu, Haofei Xu, Li Xueting, Marc Pollefeys, Ming-Hsuan Yang, and Peng Songyou. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. *arXiv preprint arXiv:2410.24207*, 2024a.

Jianglong Ye, Peng Wang, Kejie Li, Yichun Shi, and Heng Wang. Consistent-1-to-3: Consistent image to 3d view synthesis via geometry-aware diffusion models. In *2024 International Conference on 3D Vision (3DV)*, pp. 664–674. IEEE, 2024b.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4578–4587, 2021.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. *arXiv preprint arXiv:2511.20645*, 2025.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Forrester Cole, Deqing Sun, Ming-Hsuan Yang, et al. Monst3r: A simple approach for estimating geometry in the presence of motion. In *International Conference on Learning Representations*, volume 2025, pp. 82863–82886, 2025.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. Graph.*, 37(4):65, 2018. doi: 10.1145/3197517.3201323. URL https://doi.org/10.1145/3197517.3201323.

Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12588–12597, 2023.

A Related Work

Feed-forward 3D prediction. Feed-forward 3D systems replace per-scene optimization with direct prediction from images. Early generalizable radiance-field and image-based rendering models predict neural scene representations or target-view radiance from sparse inputs (Yu et al., 2021; Wang et al., 2021; Chen et al., 2021). Recent Gaussian and transformer models further scale this direct-prediction paradigm to sparse-view rendering, view synthesis, cameras, depth, pointmaps, and tracks (Charatan et al., 2024; Chen et al., 2024; Jin et al., 2025; Jiang et al., 2025; Wang et al., 2025a; 2024; Leroy et al., 2024; Wang et al., 2025b). These methods establish feed-forward prediction as a strong default for reconstruction. We place this default within a unified flow formulation to explain its successes, its failures under ambiguity, and its relationship to conditional generation.

Denoising targets. Diffusion and score-based models generate samples through iterative denoising (Ho et al., 2020; Song et al., 2020a;b; Karras et al., 2022), and flow matching gives a related continuous transport formulation (Lipman et al., 2022). Modern architectures such as diffusion transformers make the denoiser itself increasingly general-purpose (Peebles & Xie, 2023; Yu et al., 2025). The denoising target is also a modeling choice: denoisers may predict noise, velocity, scores, or clean data. Recent pixel-space work revives direct clean-target x_0 prediction (Li & He, 2025; Yu et al., 2025). Our formulation uses the noise-to-data flow-matching path throughout: clean-target prediction induces its velocity field, connecting the feed-forward endpoint with multi-step generative inference.

Generative and hybrid 3D. Generative models have been used to predict ambiguous 3D signals, including novel views, 3D structure, and depth (Watson et al., 2022; Liu et al., 2023; Chan et al., 2023; Ye et al., 2024b; Kani et al., 2023; Ke et al., 2024). Diffusion priors have also been lifted into 3D reconstruction and generation through distillation, optimization, or reconstruction regularization (Poole et al., 2022; Zhou & Tulsiani, 2023; Wu et al., 2024). More recent systems combine reconstruction and generation priors with tokenized, autoregressive, or diffusion components (Xiang et al., 2026; Chang et al., 2025; Le et al., 2025; Asim et al., 2026). These works, together with recent spatial world models such as Atlas (World Labs Team, 2026), show the practical relevance of bringing reconstruction and generation together. We study their shared formulation: reconstruction and generation emerge from different denoising configurations of one flow-based prediction interface, instantiated in both NVS and geometry.

Uncertainty and posterior means. The theory of squared-error prediction and Bayesian decision making identifies posterior or conditional means as optimal point estimates (Berger, 1985; Hastie et al., 2009). Posterior-mean denoising also has a classical score-based characterization through Tweedie's formula (Efron, 2011). Diffusion inverse-problem methods further separate posterior-mean restoration from posterior sampling (Kawar et al., 2022; Chung et al., 2022). The perception–distortion literature formalizes why mean-like estimates and perceptually realistic samples can occupy different points on a trade-off curve (Blau & Michaeli, 2018; 2019; Freirich et al., 2021). We connect these established statistical ideas to a unified empirical account of 3D prediction. The toy experiment makes conditional averaging visible, and the NVS and geometry experiments show why pointwise distortion alone can miss the benefits of flow inference.

B Flow Formulation: Statements and Proofs

We use the path and clean-target interface of Section 2, with ϵ independent of (X_0, C). Conditional distributions and risks are assumed well defined, with measurable optimal actions in an unrestricted prediction class. Squared-error claims assume finite second moments. Endpoint results allow general task losses; exact flow transport uses squared error.

B.1 Endpoint Prediction and Conditional Risk

For positive weights $w(t)$ and $t < 1$, flow matching is a weighted clean-target regression:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}\big[w(t)\|v_\theta(X_t, t, C) - (X_0 - \epsilon)\|_2^2\big] = \mathbb{E}\left[\frac{w(t)}{(1-t)^2}\|f_\theta(X_t, t, C) - X_0\|_2^2\right]. \quad (4)$$

More generally, let a include the clean prediction and any auxiliary outputs, and let ℓ_t be their complete loss. The aggregate risk and its pointwise optimal actions are

$$\mathcal{R}_\nu(a) := \mathbb{E}_{t\sim\nu,(X_0,C),\epsilon}[\ell_t(X_0, a(X_t, t, C))], \tag{5}$$

$$a_t^\star(x, c) \in \mathcal{B}_t(x, c) := \arg\min_a \mathbb{E}[\ell_t(X_0, a) \mid X_t = x, C = c], \tag{6}$$

where t is independent of the other variables. Write q_t for the clean-target readout of $a_t^\star$.

Proposition 1 (Prediction from an uninformative state)**.** *Define*

$$\mathcal{B}_\ell(c) := \arg\min_a \mathbb{E}[\ell(X_0, a) \mid C = c]. \tag{7}$$

If $S \perp X_0 \mid C$, then almost surely

$$\arg\min_a \mathbb{E}[\ell(X_0, a) \mid S = s, C = c] = \mathcal{B}_\ell(c). \tag{8}$$

A unique optimum ignores S; otherwise S can only select among equally optimal actions.

Proof. Conditional independence gives, for every admissible a,

$$\mathbb{E}[\ell(X_0, a) \mid S = s, C = c] = \mathbb{E}[\ell(X_0, a) \mid C = c]. \tag{9}$$

The conditional risks, and hence their minimizers, coincide. □

Proposition 2 (When an optimal one-step predictor cannot sample)**.** *Let π_X extract the clean prediction and set $\mathcal{Q}_\ell(c) = \{\pi_X(a) : a \in \mathcal{B}_\ell(c)\}$. Under Proposition 1, every optimal predictor's conditional output law is supported on $\overline{\mathcal{Q}_\ell(c)}$. It cannot match a target law assigning positive mass outside this set. In particular, a unique clean readout cannot sample a non-degenerate target law.*

Proof. An optimal action lies in $\mathcal{B}_\ell(c)$ almost surely, so its clean readout lies in $\mathcal{Q}_\ell(c)$. □

Reconstruction endpoints. LVSM's fixed target-side input and Noised-LVSM's independent noise both satisfy Proposition 1. Their unrestricted optima therefore agree under the same image loss. Likewise, for a matched complete geometry loss,

$$\underbrace{\arg\min_a \mathbb{E}[\ell_{\mathrm{geo}}(X_0, a) \mid \epsilon = e, C = c]}_{\text{JUSt3R endpoint}} = \underbrace{\arg\min_a \mathbb{E}[\ell_{\mathrm{geo}}(X_0, a) \mid C = c]}_{\text{DUSt3R endpoint}}. \tag{10}$$

The action may include confidence and coupled scale normalization. This identity requires the same loss on both sides; it does not equate a norm-loss optimum with a squared-error flow optimum.

For a continuous timestep distribution, $t = 0$ has zero mass in Eq. (5). The endpoint statements concern its conditional-risk minimizer. They describe the trained endpoint when it receives positive training mass, or when the predictor and its conditional optimum extend continuously to $t = 0$.

B.2 One-Step Prediction and Denoising Depth

The clean readout induces a velocity and Euler update

$$\widetilde{v}_t(x, c) = \frac{q_t(x, c) - x}{1 - t}, \tag{11}$$

$$\widetilde{X}_{t_{j+1}} = \frac{1 - t_{j+1}}{1 - t_j}\widetilde{X}_{t_j} + \frac{t_{j+1} - t_j}{1 - t_j} q_{t_j}(\widetilde{X}_{t_j}, C), \tag{12}$$

for $0 = t_0 < \cdots < t_K = 1$ and $\widetilde{X}_{t_0} = \epsilon$.

Proposition 3 (One-step Euler endpoint)**.** *One Euler step returns $\widetilde{X}_1 = q_0(\epsilon, C)$. For an optimal endpoint action, this lies in $\mathcal{Q}_{\ell_0}(C)$. A unique clean readout is independent of ϵ given C; under squared error, it is $\mathbb{E}[X_0 \mid C]$.*

Proof. Set $t_0 = 0, t_1 = 1$ in Eq. (12), then apply Proposition 1. □

For $0 < t_k < 1$, the same recurrence unrolls to

$$\widetilde{X}_{t_k} = (1 - t_k)\epsilon + t_k \bar{q}_k, \qquad \bar{q}_k = \sum_{j<k} \lambda_j^{(k)} q_{t_j}(\widetilde{X}_{t_j}, C),$$
$$\lambda_j^{(k)} = \frac{1 - t_k}{t_k} \frac{t_{j+1} - t_j}{(1 - t_j)(1 - t_{j+1})}. \tag{13}$$

The nonnegative weights sum to one by telescoping. Unlike a training state, this state contains predictions in place of the true target.

Proposition 4 (State dependence beyond one step)**.** *If $q_0(C)$ is unique, a two-step schedule $0 < s < 1$ gives $\widetilde{X}_s = (1 - s)\epsilon + s q_0(C)$ and $\widetilde{X}_1 = q_s(\widetilde{X}_s, C)$. For non-degenerate noise, the intermediate state is non-degenerate given C; the output varies exactly when $q_s(\cdot, C)$ is not almost surely constant under that state's law.*

Proof. Apply Eq. (12) twice. The first update retains the nonzero noise coefficient $1 - s$; the second evaluates q_s on this state. □

Additional steps thus permit state-dependent predictions. Exact sampling further requires the appropriate velocity and integration, as follows.

B.3 Ideal Conditional Flow Transport

Fix $C = c$, write $\rho_t = \mathcal{L}(X_t \mid C = c)$, and set $U_t = X_0 - \epsilon$. Define

$$v_t^\star(x, c) = \mathbb{E}[U_t \mid X_t = x, C = c], \qquad m_t(x, c) = \mathbb{E}[X_0 \mid X_t = x, C = c]. \tag{14}$$

The squared-error optimum is $v_t^\star$. Since $X_0 - \epsilon = (X_0 - X_t)/(1 - t)$,

$$v_t^\star(x, c) = \frac{m_t(x, c) - x}{1 - t}, \qquad t < 1. \tag{15}$$

Proposition 5 (Transport by the conditional mean field)**.** *Assume finite first moments. Then ρ_t satisfies the weak continuity equation*

$$\partial_t \rho_t + \nabla \cdot (\rho_t v_t^\star) = 0, \qquad \rho_0 = p_0. \tag{16}$$

If, on each $[0, T]$ with $T < 1$, the ODE

$$\frac{\mathrm{d}Z_t}{\mathrm{d}t} = v_t^\star(Z_t, c), \qquad Z_0 \sim p_0 \tag{17}$$

admits a unique flow and its continuity equation has a unique probability-measure solution, then $\mathcal{L}(Z_t \mid c) = \rho_t$ for $t < 1$. These laws converge weakly to $p_{\mathrm{data}}(\cdot \mid c)$ as $t \uparrow 1$; any almost-sure terminal limit Z_1 has that target law.

Proof. For a smooth compactly supported test function φ, differentiation and conditional expectation give

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[\varphi(X_t) \mid c] = \mathbb{E}[\nabla\varphi(X_t)^\top U_t \mid c] = \int \nabla\varphi(x)^\top v_t^\star(x, c)\, \rho_t(\mathrm{d}x). \tag{18}$$

Thus the interpolation and ODE marginals solve the same continuity equation with the same initial law. Uniqueness identifies them; $X_t \to X_0$ gives the terminal marginal limit. □

For a different clean readout, the velocity discrepancy is exactly

$$\widetilde{v}_t(x, c) - v_t^\star(x, c) = \frac{q_t(x, c) - m_t(x, c)}{1 - t}. \tag{19}$$

Squared-error flow matching gives $q_t = m_t$ at its population optimum. A general task loss need not, and convergence of clean predictions alone does not control the ratio near $t = 1$.

Geometry normalization. JUSt3R forms its training path using a jointly normalized two-view pointmap, a fixed representative of the target modulo scale. Intermediate sampler updates instead use per-view normalization by default, with the final prediction returned at raw scale. These projections preserve the endpoint risk argument but distinguish the practical sampler from the ideal ODE above.

B.4 The Value of Additional Context

Proposition 6 (Value of information)**.** *If $C_{\text{sparse}} = g(C_{\text{rich}})$ and both settings use the same target, action space, and loss, then*

$$\underbrace{\inf_{\delta_{\text{rich}}} \mathbb{E}[\ell(X_0, \delta_{\text{rich}}(C_{\text{rich}}))]}_{R^\star_{\text{rich}}} \leq \underbrace{\inf_{\delta_{\text{sparse}}} \mathbb{E}[\ell(X_0, \delta_{\text{sparse}}(C_{\text{sparse}}))]}_{R^\star_{\text{sparse}}}. \tag{20}$$

Proof. Every sparse-context rule is available with rich context by choosing $\delta_{\text{rich}} = \delta_{\text{sparse}} \circ g$. Taking infima proves the inequality. □

This orders expected optimal risks. It does not imply narrower posteriors for every observation or monotonic scores across empirical co-visibility bins.

C Toy Experiment Details

The toy experiment uses a known conditional distribution, so the conditional mean, the posterior over modes, and the correct latent mode are all measurable. This appendix records the data distribution, the compared predictors, the flow model and its sampler, the evaluation protocol, and full results for Section 3.1. The key design choice is that a single flow checkpoint per reliability and seed is evaluated at every step count, so one-step prediction and multi-step sampling are two inference configurations of the same model.

Data distribution. We sample a latent mode:

$$z \sim \text{Uniform}(\{0, \dots, K-1\}). \tag{21}$$

In the main experiment, $K = 8$ mode centers are placed on a circle of radius 2:

$$\mu_k = 2\,(\cos(2\pi k/K),\, \sin(2\pi k/K))\,. \tag{22}$$

The target point is sampled as:

$$x_0 \mid z = k \sim \mathcal{N}(\mu_k, \sigma_m^2 I). \tag{23}$$

We use $\sigma_m = 0.15$. The context is a noisy one-hot observation of the latent mode:

$$c_r \mid z = k \sim \mathcal{N}(e_k, \sigma_{\text{ctx}}(r)^2 I), \qquad r \in [0, 1]. \tag{24}$$

Here e_k is the one-hot vector for mode k.

$$\sigma_{\text{ctx}}(r) = \sigma_{\text{sparse}} + r\,(\sigma_{\text{rich}} - \sigma_{\text{sparse}})\,, \qquad \sigma_{\text{sparse}} = 2.5, \quad \sigma_{\text{rich}} = 0.08. \tag{25}$$

The reliability parameter r controls the context noise and hence the posterior concentration over modes, while leaving the marginal target distribution unchanged.

Analytic references. Because the context likelihood is Gaussian around one-hot vertices, the posterior over modes can be computed analytically:

$$p(z = k \mid c_r) = \frac{\exp\!\big(-\|c_r - e_k\|_2^2/(2\sigma_{\text{ctx}}(r)^2)\big)}{\sum_j \exp(-\|c_r - e_j\|_2^2/(2\sigma_{\text{ctx}}(r)^2))}. \tag{26}$$

The analytic conditional mean is:

$$\mathbb{E}[x_0 \mid c_r] = \sum_{k=0}^{K-1} p(z = k \mid c_r)\mu_k. \tag{27}$$

This is the reference plotted as `mean` in Figure 1. We also use an exact posterior sampler, `exact`, which draws $z \sim p(z \mid c_r)$ from Eq. (26) and then $x_0 \sim \mathcal{N}(\mu_z, \sigma_m^2 I)$. It is the ideal behavior of a conditional generative model and serves as the reference for multi-step sampling.

Compared predictors. `gt` is sampled from the true target distribution. `feed-forward` is a deterministic predictor $f(c_r) \to x_0$ trained with $\mathbb{E}\|f(c_r) - x_0\|_2^2$. The flow model is a clean-target predictor $f_\theta(x_t, t, c_r)$ on the noise-to-data path $x_t = (1-t)\epsilon + tx_0$ of Section 2, with $\epsilon \sim \mathcal{N}(0, I)$. It is trained with the clean-target form of the flow-matching loss,

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\big[w(t)\,\|f_\theta(x_t, t, c_r) - x_0\|_2^2\big]\,, \qquad w(t) = \frac{1}{(1 - \min(t, 0.95))^2}, \tag{28}$$

which equals the squared error of the induced velocity $(f_\theta - x_t)/(1-t)$ against $x_0 - \epsilon$. Time is sampled as $t = 0$ with probability 0.1 and $t \sim \mathcal{U}[0, 0.95]$ otherwise, so the one-step endpoint receives positive training mass rather than being reached by extrapolation.

Both learned models use an MLP trunk of hidden width 256 and depth 4, AdamW with learning rate 10^{-3}, weight decay 10^{-4}, and batch size 512, and one model of each kind is trained per seed and reliability. In the flow model, the context (a 2-layer MLP) and time (128 sinusoidal features of $1000t$ followed by a 2-layer MLP) modulate every hidden layer through zero-initialized FiLM, and the raw context is also concatenated to the input. This conditioning pathway brings the flow model to 1.09M parameters, against 0.13M for `feed-forward`. `feed-forward` is trained for $12{,}000$ steps and the flow model for $36{,}000$; longer training (72k steps) and a wider trunk (512) gave no further improvement.

Sampling. We integrate the induced velocity with Euler steps on the uniform grid $t_i = i/k$, $i = 0, \ldots, k$, starting from $x_{t_0} = \epsilon$. The final step lands exactly on the clean prediction, so $k = 1$ returns $f_\theta(\epsilon, 0, c_r)$, the one-step endpoint. The network's time input is clamped to its training range ($t \le 0.95$), which affects only schedules with $k > 20$. All step counts use the same checkpoint; `flow-1` and `flow-15` in Figure 1 denote $k = 1$ and $k = 15$.

Evaluation protocol. The main sweep uses $r = 0.0, 0.1, \ldots, 1.0$ and five random seeds. Metrics are evaluated with 16,384 samples per seed for $k = 1$ and $k = 15$, and with 4,096 samples per seed for the step sweep $k \in \{1, 2, 3, 5, 10, 15, 25, 50\}$. For a predicted point $\hat{x}$, we report per-coordinate distortion $\mathbb{E}[\|\hat{x} - x_0\|_2^2/2]$, per-coordinate conditional-mean error $\mathbb{E}[\|\hat{x} - \mathbb{E}[x_0 \mid c_r]\|_2^2/2]$, nearest-mode distance $\mathbb{E}[\min_k \|\hat{x} - \mu_k\|_2]$, and mode consistency $\Pr[\arg\min_k \|\hat{x} - \mu_k\|_2 = z]$. Diversity is the mean trace of the empirical covariance of eight samples drawn with independent noise for the same context; `feed-forward` is deterministic, so its diversity is 0 by construction. For `exact`, distortion and diversity are both close to twice its conditional-mean error, as expected for a posterior sampler.

Quantitative results. Table 4 reports the reliabilities shown in Figure 1 together with $r = 0$, as mean$\pm$std over five seeds. Low reliability tests whether a predictor follows the posterior mean or samples from the data manifold; high reliability tests whether sampling becomes mode-consistent once the context identifies the latent mode.

One-step endpoint. Across all eleven reliabilities, the conditional-mean error of `flow-1` is at most 0.009 (mean over seeds), the same order as `feed-forward` (at most 0.0044) and two orders of magnitude below any posterior sampler. At $r = 0.0$ and $r = 0.4$, both one-step predictors lie far from the mode centers (nearest-mode distance above 1.5, against 0.19 for data): the conditional mean sits between the modes.

Multi-step sampling. From the same checkpoint, `flow-15` returns to the data manifold and matches the mode consistency of `exact` at every reliability: it is at chance when the context is ambiguous (0.142 against 0.146 at $r = 0$), and identical to `exact` at $r = 0.9$ (0.896) and $r = 1.0$ (1.000). Table 5 repeats this comparison on a finer grid near rich context, where the posterior contracts fastest; `flow-15` follows the exact curve throughout the transition.

Within-mode dispersion. The one systematic gap is within-mode spread at high reliability: at $r = 1.0$, `flow-15` places every sample on the correct mode but with nearest-mode distance 0.157 and diversity 0.028, against 0.188 and 0.045 for `exact`. To separate learning error from sampling error, we run the same Euler sampler with the Bayes-optimal clean-target predictor $\mathbb{E}[x_0 \mid x_t, c_r]$, which is available in closed form for this Gaussian mixture. It shows the same shortfall (nearest-mode distance 0.151 and diversity 0.029 at $k = 15$), because the within-mode spread ($\sigma_m = 0.15$) is determined at $t \gtrsim 0.85$, where a uniform 15-step grid takes only one or two steps. The gap therefore

Table 4: Toy metrics, mean±std over five seeds. One flow checkpoint per (r, seed) is evaluated with one Euler step (`flow-1`) and with 15 (`flow-15`). `mean` is the analytic $\mathbb{E}[x_0 \mid c]$ and `exact` the exact posterior sampler. One-step predictors match `mean`; the 15-step sampler matches the mode selection of `exact`. CM error: per-coordinate squared error to $\mathbb{E}[x_0 \mid c]$; mode dist.: distance to the nearest mode center; mode cons.: fraction of predictions nearest the true mode.

Reliability	Method	CM error ↓	Mode dist. ↓	Mode cons. ↑	Distortion ↓	Diversity
$r = 0.0$	`exact`	1.98173±0.00244	0.18787±0.00069	0.14550±0.00360	3.95202±0.02895	3.95706±0.00373
$r = 0.0$	`mean`	0.00000±0.00000	1.75274±0.00107	0.16716±0.00050	1.97979±0.00252	–
$r = 0.0$	`feed-forward`	0.00136±0.00036	1.74957±0.01735	0.16368±0.00349	1.98199±0.00303	–
$r = 0.0$	`flow-1`	0.00256±0.00084	1.76830±0.01112	0.16583±0.00163	1.98313±0.00390	0.00091±0.00031
$r = 0.0$	`flow-15`	1.86915±0.05846	0.19175±0.01137	0.14235±0.00379	3.85215±0.05699	3.72978±0.11580
$r = 0.4$	`exact`	1.90867±0.00603	0.18853±0.00102	0.17390±0.00179	3.83301±0.02287	3.80880±0.00182
$r = 0.4$	`mean`	0.00000±0.00000	1.58902±0.00148	0.20696±0.00241	1.90754±0.00181	–
$r = 0.4$	`feed-forward`	0.00218±0.00046	1.59005±0.01717	0.20934±0.00445	1.90542±0.00610	–
$r = 0.4$	`flow-1`	0.00424±0.00175	1.60486±0.01325	0.20835±0.00383	1.90896±0.00876	0.00165±0.00029
$r = 0.4$	`flow-15`	1.79413±0.03077	0.18713±0.00657	0.17341±0.00172	3.69475±0.03906	3.57342±0.06062
$r = 0.9$	`exact`	0.25936±0.00326	0.18791±0.00071	0.89568±0.00192	0.51804±0.00750	0.52255±0.00980
$r = 0.9$	`mean`	0.00000±0.00000	0.17002±0.00114	0.90859±0.00194	0.25670±0.00528	–
$r = 0.9$	`feed-forward`	0.00441±0.00079	0.19809±0.00603	0.90739±0.00267	0.26203±0.00810	–
$r = 0.9$	`flow-1`	0.00788±0.00272	0.20096±0.01277	0.90518±0.00172	0.27021±0.00454	0.00399±0.00053
$r = 0.9$	`flow-15`	0.23120±0.00654	0.16251±0.00685	0.89617±0.00390	0.48006±0.01744	0.45125±0.01516
$r = 1.0$	`exact`	0.02261±0.00021	0.18841±0.00096	1.00000±0.00000	0.04489±0.00046	0.04504±0.00015
$r = 1.0$	`mean`	0.00000±0.00000	0.00000±0.00000	1.00000±0.00000	0.02250±0.00028	–
$r = 1.0$	`feed-forward`	0.00030±0.00014	0.02132±0.00592	1.00000±0.00000	0.02277±0.00031	–
$r = 1.0$	`flow-1`	0.00049±0.00021	0.02725±0.00697	1.00000±0.00000	0.02285±0.00024	0.00017±0.00006
$r = 1.0$	`flow-15`	0.01566±0.00102	0.15657±0.00538	0.99999±0.00003	0.03822±0.00085	0.02830±0.00198

Table 5: Mode consistency near rich context (seed 0). `flow-15` tracks the exact posterior sampler as the posterior contracts.

	$r = 0.80$	$r = 0.85$	$r = 0.90$	$r = 0.95$	$r = 1.00$
`exact`	0.510	0.689	0.898	0.998	1.000
`flow-15`	0.519	0.690	0.895	0.997	1.000

reflects discretization near the data endpoint, not a failure of the learned predictor or its conditioning. In fact, it narrows with more steps (nearest-mode distance 0.176 at $k = 50$).

Robustness to $K = 16$. With 16 modes ($r \in \{0.0, 0.5, 1.0\}$, three seeds, 8,192 samples per seed), the same behavior holds. The one-step endpoint matches the analytic mean (conditional-mean error 0.002, 0.005, and 0.0007), and at $r = 1.0$ the 15-step samples reach mode consistency 0.996, against 0.991 for `exact`; at $r = 0.0$ both are at chance.

D LVSM AND JIT-LVSM EXPERIMENT DETAILS

This section provides implementation details for the LVSM-family experiments used in the main paper. We organize the details around the experimental protocol, controlled model variants, shared backbone, JiT-LVSM objective, training recipe, inference procedure, and evaluation metrics. Unless otherwise stated, all LVSM-family variants share the same conditioning inputs and backbone shape, so that the comparisons mainly isolate the effect of target-side source distribution, training objective, and denoising depth.

Datasets, protocols, and context-richness proxies. The real-NVS experiments are conducted on RealEstate10K (Zhou et al., 2018) and DL3DV (Ling et al., 2024). We evaluate both interpolation and extrapolation protocols. In the interpolation protocol, target views remain close to the observed view span. In the extrapolation protocol, target views are less directly supported by the source images and therefore expose ambiguity more strongly. The main results report sparse- and rich-context splits and a source-overlap probe based on RoMA (Edstedt et al., 2024). These characterize the evidence available to constrain a target view; they are practical context descriptors rather than measurements of conditional entropy.

Controlled model variants. We compare three LVSM-family variants. The LVSM baseline is the original one-step feed-forward reconstruction model. It uses source RGB and Plucker-ray tokens together with target Plucker-ray tokens, and directly predicts target RGB patches. Noised-LVSM changes only the target-side source distribution: it replaces blank target-side RGB channels with Gaussian noise while keeping the same one-step x_0 prediction objective. JiT-LVSM instantiates the shared clean-target flow interface with the LVSM backbone and conditioning inputs. It trains on intermediate noise-to-data states with a flow-matching objective and image-space supervision. During training, the target-side image input is a time-dependent mixture of Gaussian noise and the ground-truth target image. During inference, JiT-LVSM starts from noise and synthesizes the target view through multi-step denoising.

Shared backbone architecture. Unless otherwise stated, LVSM, Noised-LVSM, and JiT-LVSM use the same decoder-only transformer backbone. Images are resized to 224×224 and tokenized with 14×14 patches, giving a 16×16 patch grid per view. Each image token has 9 input channels: RGB plus two 3-channel ray or camera-direction representations. The transformer has hidden width 768, head dimension 64, and 12 layers. We use QK normalization, rotary positional embeddings with frequency 100, VGGT-style attention, special initialization, and depth initialization.

JiT-LVSM objective and time parameterization. JiT-LVSM follows the clean-data prediction interface of JiT while preserving the LVSM conditioning interface. For each target image x_0, we sample Gaussian noise ϵ and a flow-matching time t uniformly along the noise-to-data path. The target-side input is the linear interpolation $x_t = tx_0 + (1-t)\epsilon$. The network predicts $\hat{x}_0$ from source-view tokens, target-ray tokens, x_t, and a timestep embedding injected into the transformer blocks. Flow matching is applied through the induced velocity, alongside an image-space task loss on the clean prediction. We describe these components as the v-loss and x-loss, respectively. The x-loss directly supervises the clean prediction $\hat{x}_0$ with a pixel MSE term and a perceptual term. The recorded representative configuration uses a pixel MSE weight of 1.0, a perceptual-loss weight of 0.5, and an LPIPS-loss weight of 0.0. The v-loss applies flow matching to the velocity induced by the predicted clean image, following the clean-target parameterization of JiT. Specifically, the target velocity is $(x_0 - x_t)/(1-t)$ and the predicted velocity is $(\hat{x}_0 - x_t)/(1-t)$. This velocity parameterization can become unstable near the data endpoint, so we follow the JiT-style stabilization and use an endpoint cutoff $t_\epsilon = 0.05$. The representative configuration weights this v-loss by 0.1. Training timesteps follow the standard uniform flow-matching schedule with this endpoint cutoff. Both terms act through the same clean-target prediction. As Eq. (4) shows, the squared velocity loss itself induces a time-dependent weighting of clean-target error; its relative contribution cannot be read from the scalar loss weights alone.

Training setup. The representative JiT-LVSM run uses RealEstate10K full-resolution processed training and test lists. Each training sample contains 2 input views and 6 target views, for 8 total views. Source views are sampled with frame distance between 25 and 192 frames. We use square center crops with scene scale factor 1.35. Training uses mixed precision with fp16 activations, TF32 enabled, and gradient checkpointing every transformer layer. The per-GPU batch size is 16 over 8 GPUs, giving a global batch size of 128 with no gradient accumulation. We train for 100k optimization steps with 3k warmup steps followed by cosine learning-rate decay. The peak learning rate is 4×10^{-4}. The optimizer uses AdamW-style moments $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.05, and gradient clipping at norm 3.0. We evaluate and checkpoint every 4k steps, and visualize 8 evaluation batches per evaluation.

Inference and compute. For multi-step JiT-LVSM evaluation and visual results, we use Heun integration with 10 denoising steps unless otherwise specified. The inference time grid follows the standard uniform flow-matching schedule from Gaussian noise to the data endpoint. The main JiT-LVSM training run used 8 A100 GPUs for approximately 20 hours.

Evaluation metrics and qualitative analysis. We report PSNR and SSIM for reference-image agreement, together with LPIPS and FID for perceptual and distributional comparison. These measurements are read alongside qualitative results; none alone fully characterizes agreement with the underlying scene. Qualitative comparisons are used to expose the main trade-off: feed-forward

outputs can be blurrier but more conservative, while multi-step denoising can produce sharper details that may or may not be scene-consistent under sparse context.

D.1 Target-Side Noise Sanity Check

Before training along the full flow, we test whether changing only the target-side input affects a one-step predictor. LVSM fills its target-side RGB channels with blanks. Noised-LVSM replaces these blanks with Gaussian noise while retaining the same backbone, context, and one-step clean-image objective. This puts the target-side state at the noise endpoint of the formulation, without adding target-specific information beyond the context.

Figure 7 isolates the effect of the target-side input: extrapolation improves with Gaussian noise even though the task remains one-step prediction. This finite-model result complements the population endpoint equivalence. An unchanged optimal-action set does not require identical optimization or generalization in trained networks.

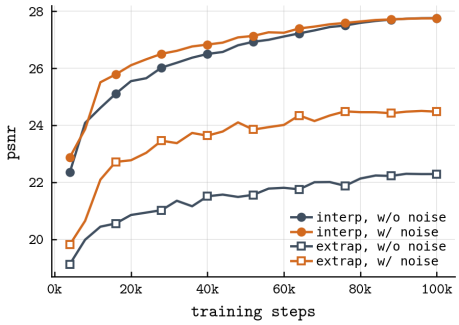

Figure 7: Noised-LVSM on RealEstate10K: extrapolation improves while interpolation remains comparable.

E DUSt3R and JUSt3R Experiment Details

This section documents the geometry instance of the unified flow formulation: the model, training recipe, evaluation protocol, and boundary metrics used in Section 3.3.

Model. JUSt3R uses DUSt3R's encoder, decoder backbone, and output heads (Wang et al., 2024). Three components are added: (i) a noisy-target embedding that patch-embeds x_t, passes it through a two-layer ViT and a zero-initialized projection, and adds the result to the decoder tokens; (ii) a sinusoidal timestep embedding with a zero-initialized MLP; and (iii) adaLN modulation of each decoder block (shift/scale/gate on self-attention, cross-attention and MLP), applied externally so that CroCo blocks are unmodified. Gates are initialized to one and shifts/scales to zero, preserving the initial feed-forward computation before the new conditioning branches are learned. Classifier-free guidance dropout is applied to the encoder features with probability 0.1 during training. The noisy-target module is a two-layer width-384 transformer, accompanied by the timestep MLP; these additions are small relative to the ViT-L encoder and twelve-layer decoder. The encoder can be shared across sampling steps. Thus k steps require repeated decoder evaluations over one context encoding, with cost also depending on the solver and guidance settings.

Geometry task loss and the endpoint analysis. DUSt3R regresses pointmaps with a confidence-weighted $L_{2,1}$ objective. Both views are expressed in camera view of view 1, write $X_i^v \in \mathbb{R}^3$ and $\bar{X}_i^v \in \mathbb{R}^3$ for the predicted and ground-truth coordinates of pixel i in view $v \in \{1,2\}$, $\mathcal{V}^v$ for its valid pixels, and $C_i^v > 0$ for the predicted confidence. The per-pixel regression term is the Euclidean distance between scale-normalized pointmaps,

$$\ell(v,i) = \left\| \tfrac{1}{z} X_i^v - \tfrac{1}{\bar{z}} \bar{X}_i^v \right\|_2, \qquad z = \frac{1}{|\mathcal{V}^1| + |\mathcal{V}^2|} \sum_v \sum_{i \in \mathcal{V}^v} \|X_i^v\|_2, \tag{29}$$

with $\bar{z}$ the same average distance computed on the ground truth, so the comparison is scale-invariant. Note that the $\|\cdot\|_2$ is taken to the first power, *not* squared. The full objective then weights each term by its confidence and rewards confident pixels through a $-\alpha \log C$ regularizer, and we get

$$\mathcal{L} = \sum_{v \in \{1,2\}} \frac{1}{|\mathcal{V}^v|} \sum_{i \in \mathcal{V}^v} \Big(C_i^v \, \ell(v,i) - \alpha \log C_i^v \Big), \qquad \alpha = 0.2. \tag{30}$$

The action in this objective contains both the pointmap and confidence, and scale normalization couples its coordinates. Consequently, its Bayes optimum is a joint action; it cannot in general

be identified with independent pixelwise geometric medians. The geometric-median characterization applies to a simpler Euclidean norm loss with fixed weights and no prediction-dependent normalization. This distinction leaves the endpoint result intact: conditional independence makes the full conditional risk the same with or without independent target-side noise (Proposition 1). The geometry experiments therefore examine the structural behavior of a direct estimator and multi-step flow predictions, without requiring DUSt3R's learned output to equal an analytic mean or median.

Flow-matching formulation. JUSt3R predicts a clean, normalized pointmap $\widehat{G}_0 = f_\theta(G_t, t, C)$ from $G_t = (1-t)\epsilon + tG_0$, where G_0 is the normalized target defined below. Its velocity is

$$v_\theta(G_t, t, C) = \frac{\widehat{G}_0 - G_t}{1-t}, \qquad U_t = G_0 - \epsilon. \tag{31}$$

The flow-matching objective supervises this velocity against U_t, using the same clean-target interface as JiT-LVSM. Equation (30) specifies the geometry task criterion used to describe the reconstruction endpoint; it is not a substitute for the complete flow-training objective. The endpoint equality in Eq. (10) holds when the complete criterion is matched.

Noise-space normalization. Pointmaps have scene-dependent scale (indoor $\sim$ 1m, outdoor $\sim$ 100m), so adding $\mathcal{N}(0, I)$ noise in raw coordinates gives inconsistent noise-to-signal ratios. We normalize the target by its average distance from the origin before forming x_t, which is the same statistic the loss uses. Therefore, the model operates in a unit-scale space and global alignment recovers metric scale downstream. This fixes the vector-space representative used by the flow interpolation; the pointmap comparison remains scale-invariant. Appendix B discusses the relation between this training normalization and intermediate sampler projections.

Training. The training of JUSt3R follows the official DUSt3R three-stage recipe without modification: stage 1 is trained at resolution of 224 with a linear head (100 epochs, 10 epochs for warmup), stage 2 is trained at resolution 512 with a linear head (100 epochs, 20 epochs for warmup), stage 3 is trained at resolution of 512 with a DPT head (90 epochs, 15 epochs for warmup); learning rate 10^{-4} throughout. JUSt3R shares the stage-1 initialization and data mixture of DUSt3R.

Evaluation protocol and metrics. Following the two-view depth protocol of MonST3R and CUT3R (Zhang et al., 2025; Wang et al., 2025c), we evaluate on five monocular-depth benchmarks unseen during training: NYU-v2 (Silberman et al., 2012), TUM (Sturm et al., 2012), Bonn (Palazzolo et al., 2019), Sintel (Butler et al., 2012), and KITTI (Geiger et al., 2013). Each image is paired with itself, and we average the z-coordinate of the first-view pointmap across the two symmetrized orderings. Before evaluation, each prediction is aligned to the ground truth using a single median scale factor. **Distortion.** We report the standard pointwise metrics AbsRel, defined as the mean of $|\hat{z} - z|/z$, and $\delta_{1.25}$, the fraction of pixels satisfying $\max(\hat{z}/z, z/\hat{z}) < 1.25$. **Geometric realism.** We additionally report three metrics that capture boundary artifacts overlooked by pointwise distortion. *Veil rate* measures predictions that fall into the empty space between two surfaces. We identify ground-truth crossings using 3×3 windows whose depth contrast satisfies $(z_{\text{far}} - z_{\text{near}})/z_{\text{near}} > 0.1$ and whose valid depths all lie within $0.25(z_{\text{far}} - z_{\text{near}})$ of either extreme, thereby excluding smooth slopes and multi-surface junctions. For each crossing, we compute

$$t = \frac{\hat{z} - z_{\text{near}}}{z_{\text{far}} - z_{\text{near}}},$$

and report the fraction for which $0.25 < t < 0.75$. *Flying-pixel rate* is computed after unprojecting the aligned depth with the dataset intrinsics. At each point p, we estimate the surface normal $n(p)$ from the covariance of its $k = 16$ nearest 3D neighbours and flag grazing geometry when

$$\left| \left\langle n(p), \frac{p}{\|p\|} \right\rangle \right| < \cos 85^\circ = 0.087,$$

following the ShadowFilter criterion (Pomerleau et al., 2013). Finally, *BF1* is the tolerance-free, scale-invariant boundary F1 of Depth Pro (Bochkovskiy et al., 2025), computed on inverse depth in four directions. Precision and recall require exact agreement between predicted and ground-truth neighbour-pair edges, without dilation or non-maximum suppression, and are averaged over ten ratio thresholds linearly spaced in $[1.05, 1.25]$, with weights proportional to the threshold. Ground-truth holes are excluded through neighbour-pair validity masks.

Additional visual comparisons. Appendix G compares depth and point clouds on NYU-v2 (Silberman et al., 2012), KITTI (Geiger et al., 2013), and BlendedMVS (Yao et al., 2020) at resolution 512. Each case places its input, reference depth, and DUSt3R/JUSt3R predictions directly above the corresponding point clouds, rendered from a 14° offset view. Wide KITTI depth panels use two rows to preserve legibility. These comparisons show how multi-step JUSt3R reduces flying pixels and separates surfaces near occlusion boundaries.

F More Visual Results for JiT-LVSM

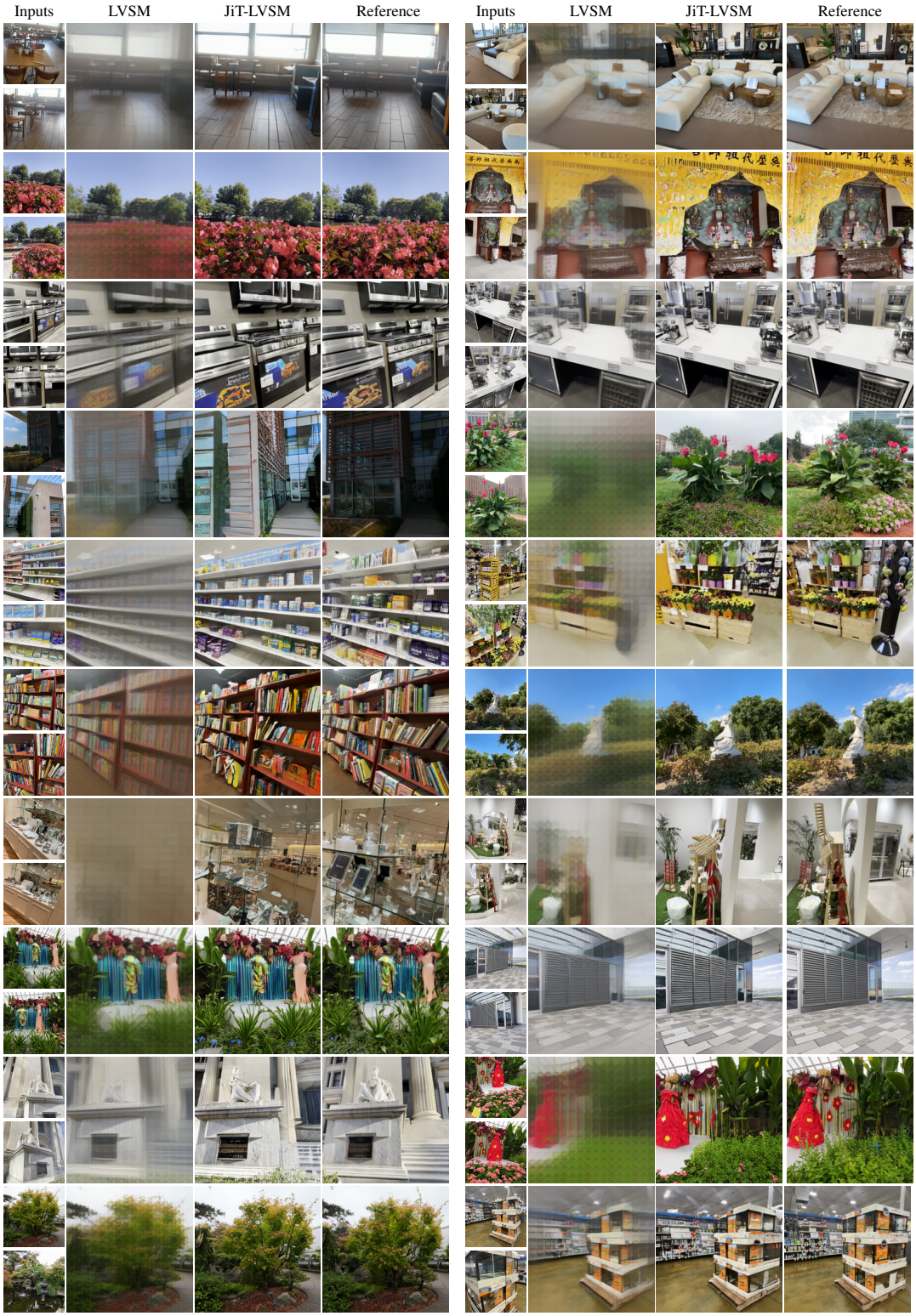

Figure 8: Random DL3DV visual samples, page 1. Each row shows two randomly selected scenes; each scene includes two input views and one target view, with columns ordered as LVSM, JiT-LVSM, and reference.

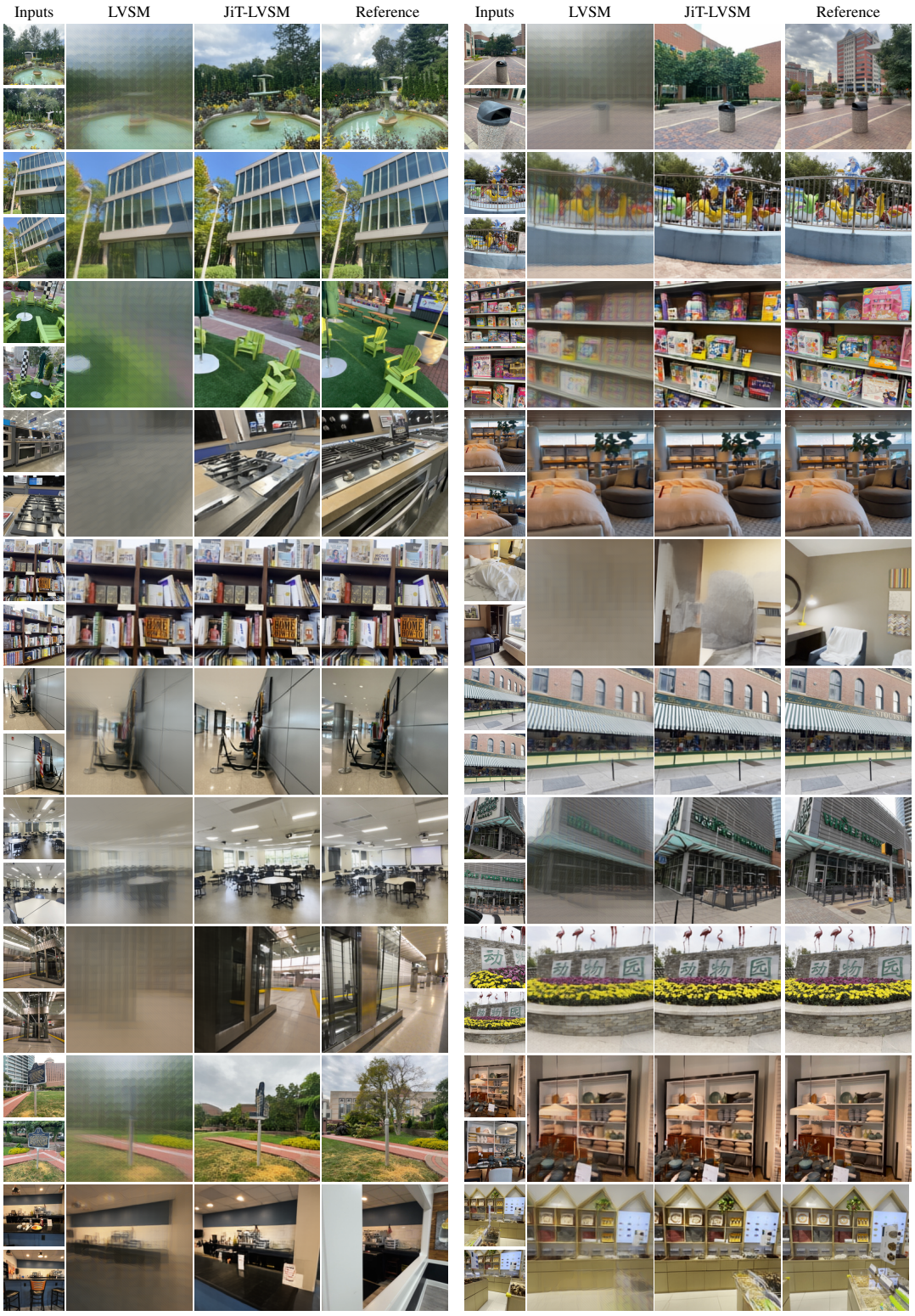

Figure 9: Random DL3DV visual samples, page 2. Each row shows two randomly selected scenes; each scene includes two input views and one target view, with columns ordered as LVSM, JiT-LVSM, and reference.

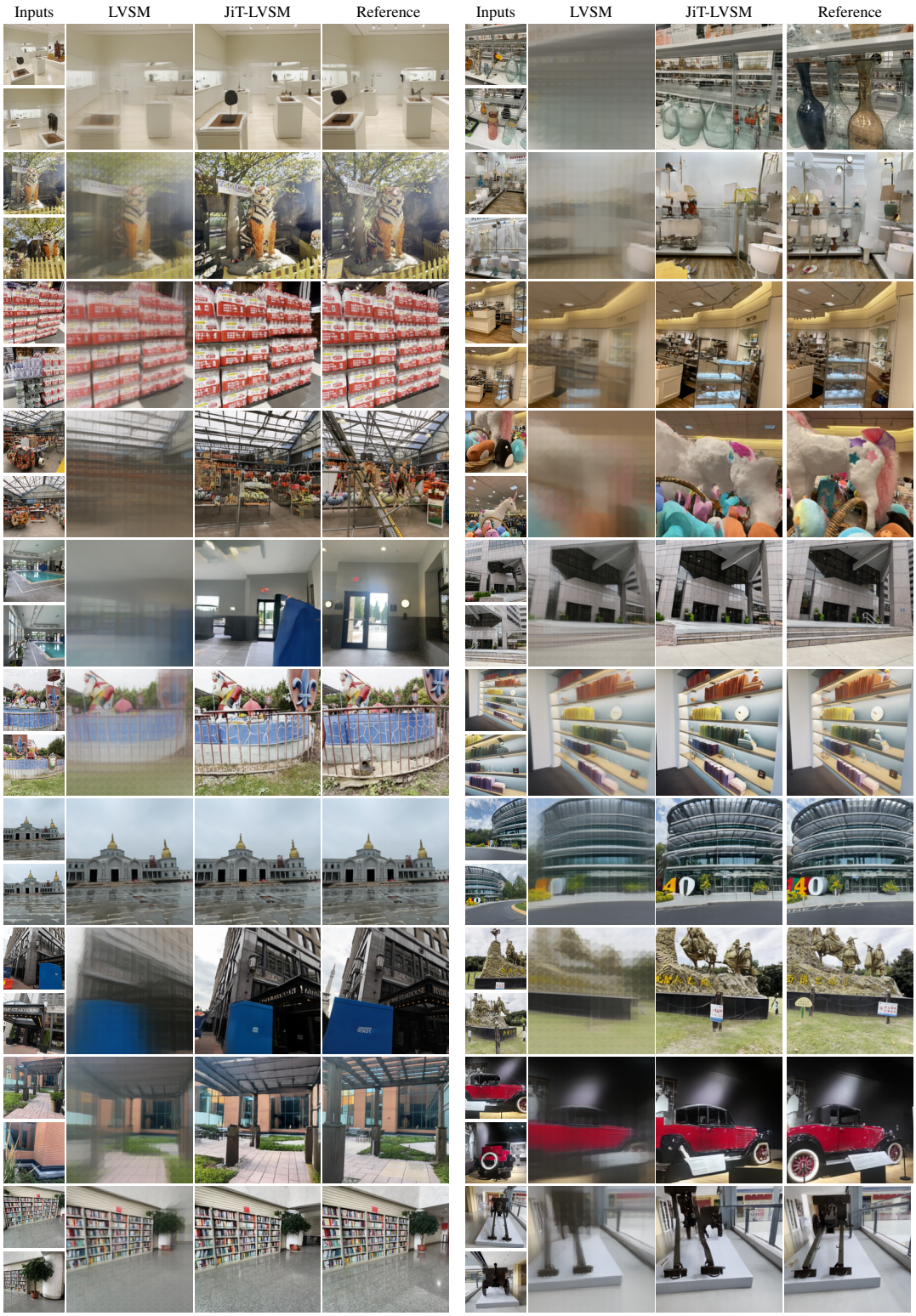

Figure 10: Random DL3DV visual samples, page 3. Each row shows two randomly selected scenes; each scene includes two input views and one target view, with columns ordered as LVSM, JiT-LVSM, and reference.

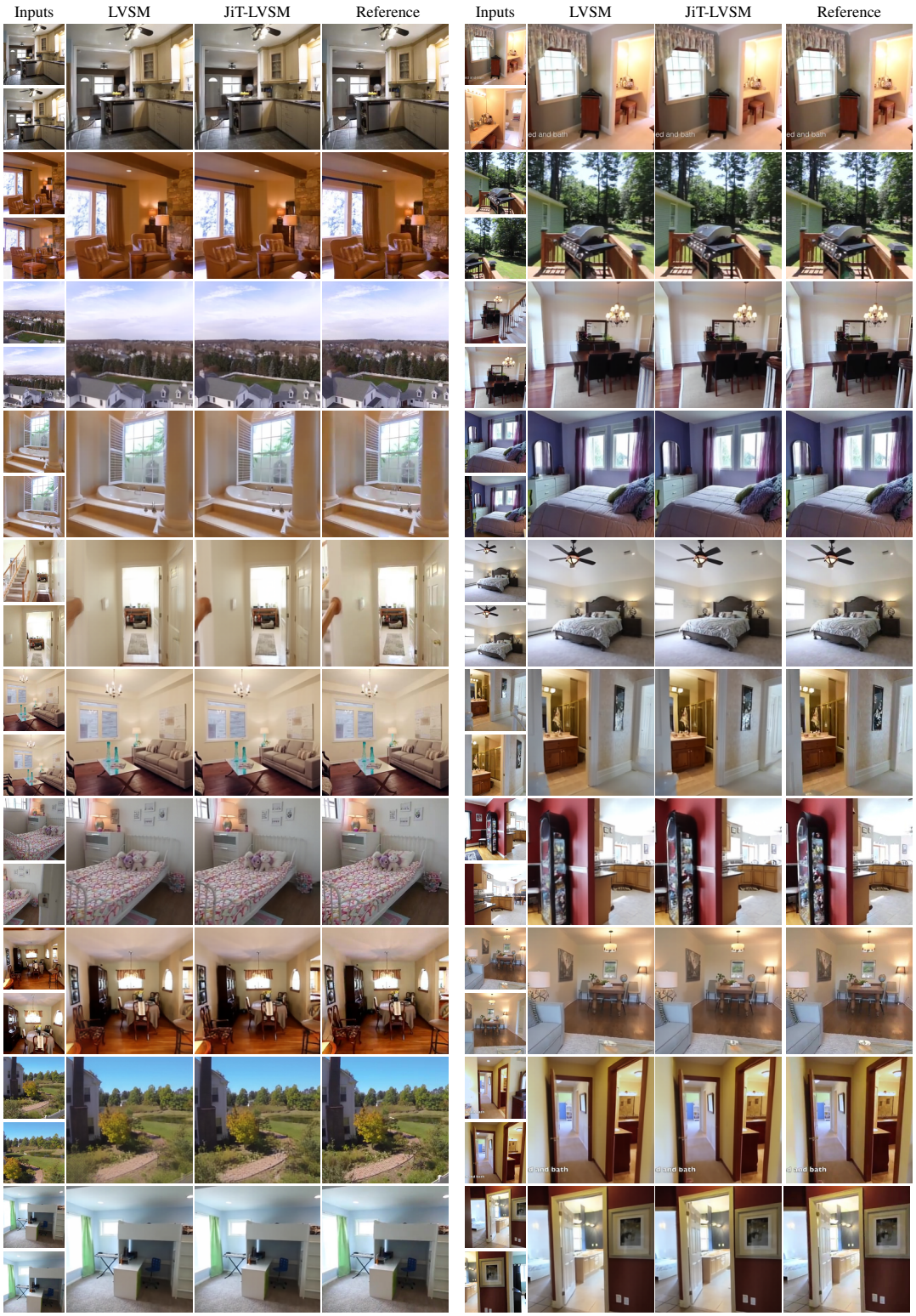

Figure 11: Random RealEstate10K visual samples, page 1. Each row shows two randomly selected scenes; each scene includes two input views and one target view, with columns ordered as LVSM, JiT-LVSM, and reference.

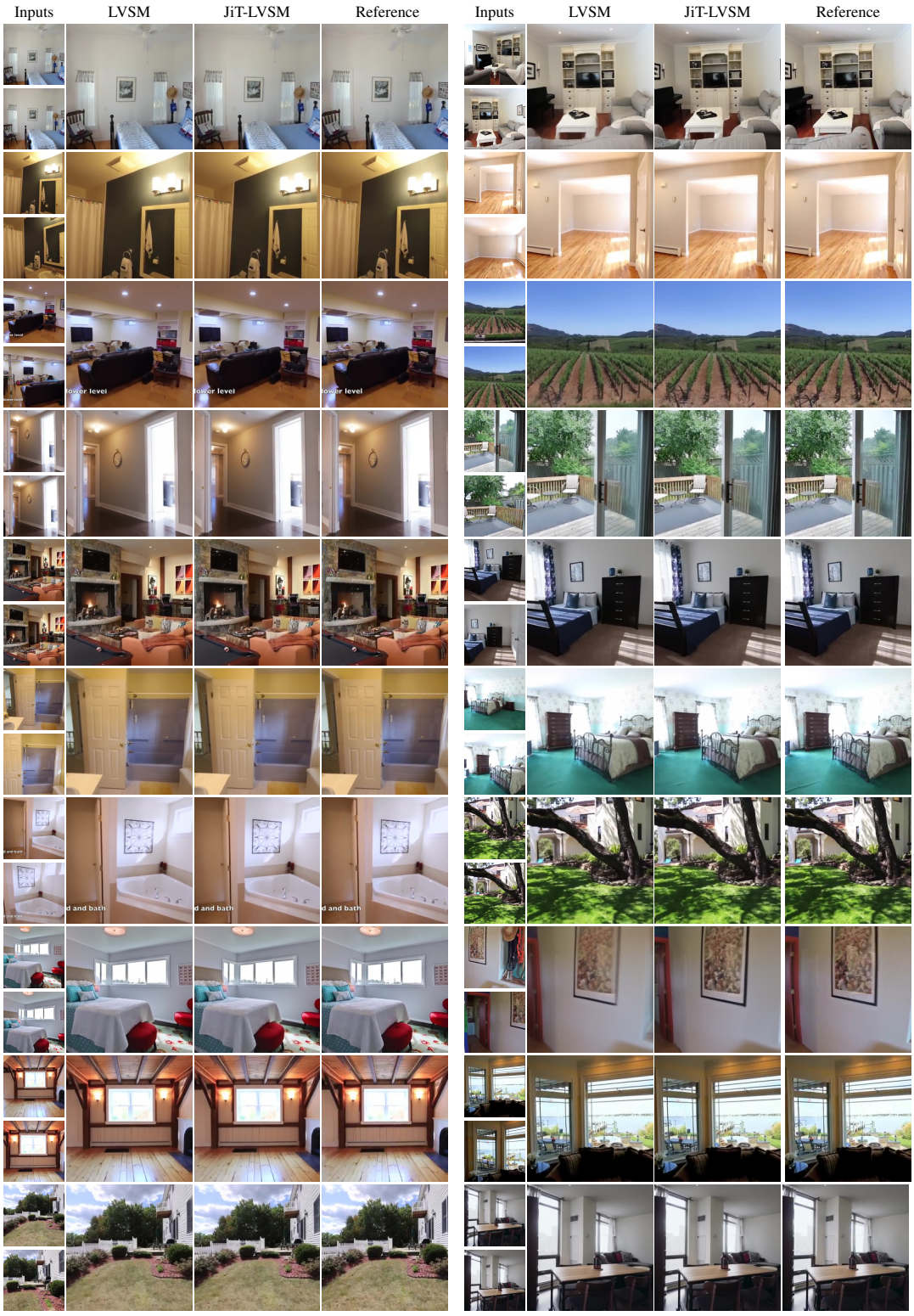

Figure 12: Random RealEstate10K visual samples, page 2. Each row shows two randomly selected scenes; each scene includes two input views and one target view, with columns ordered as LVSM, JiT-LVSM, and reference.

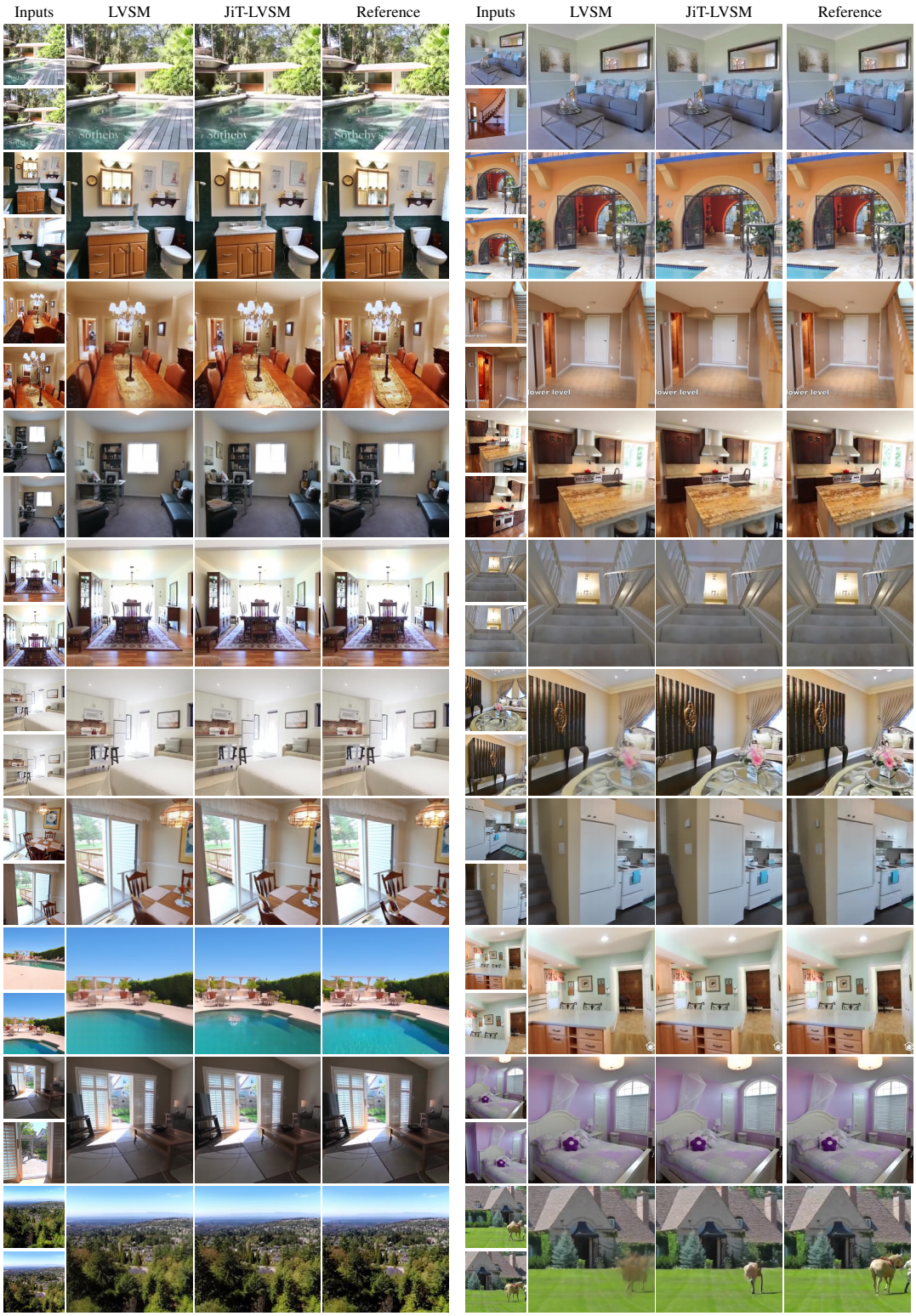

Figure 13: Random RealEstate10K visual samples, page 3. Each row shows two randomly selected scenes; each scene includes two input views and one target view, with columns ordered as LVSM, JiT-LVSM, and reference.

G MORE VISUAL RESULTS FOR JUST3R

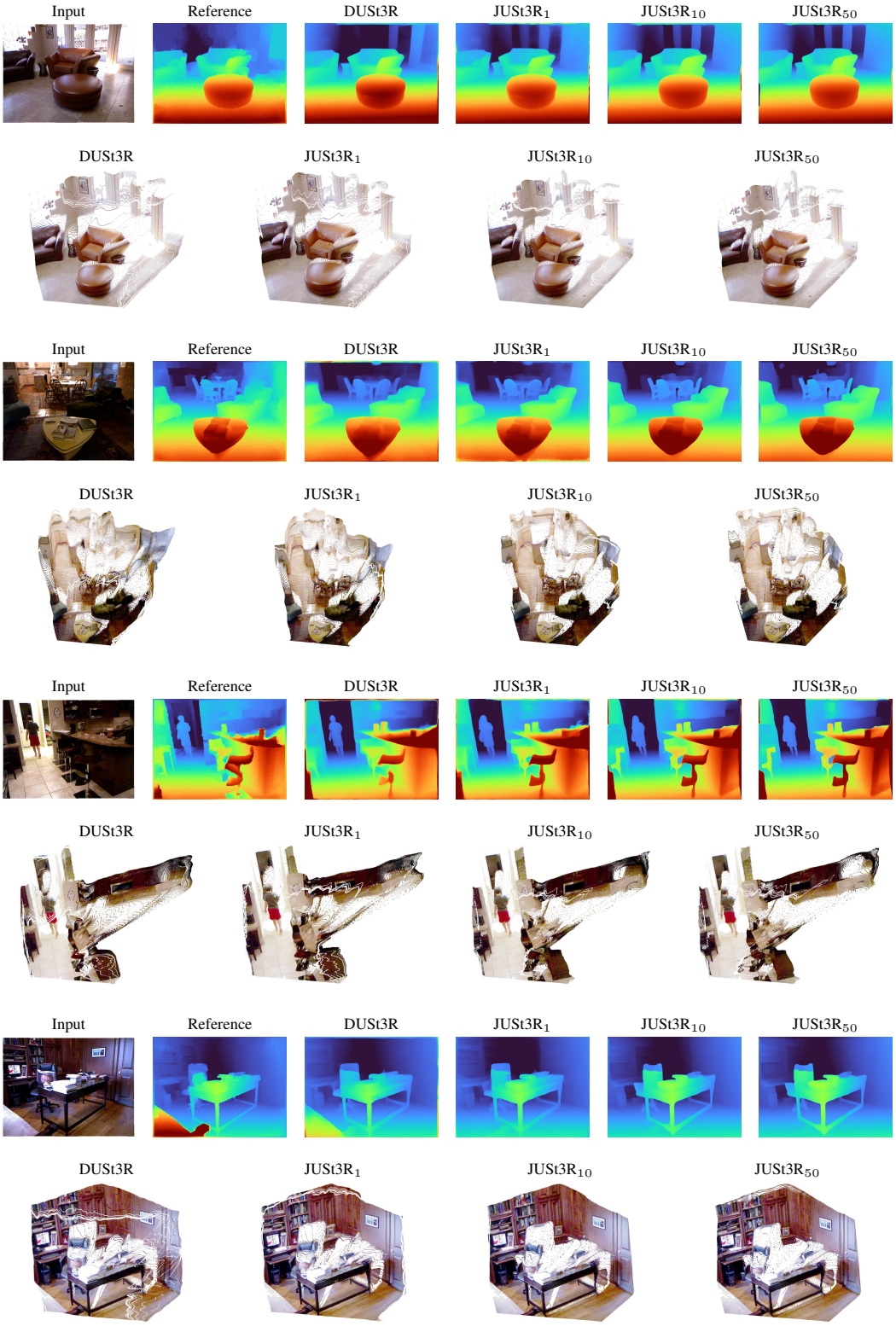

Figure 14: Additional JUSt3R comparisons, page 1. Each case shows depth predictions followed immediately by its offset point-cloud views.

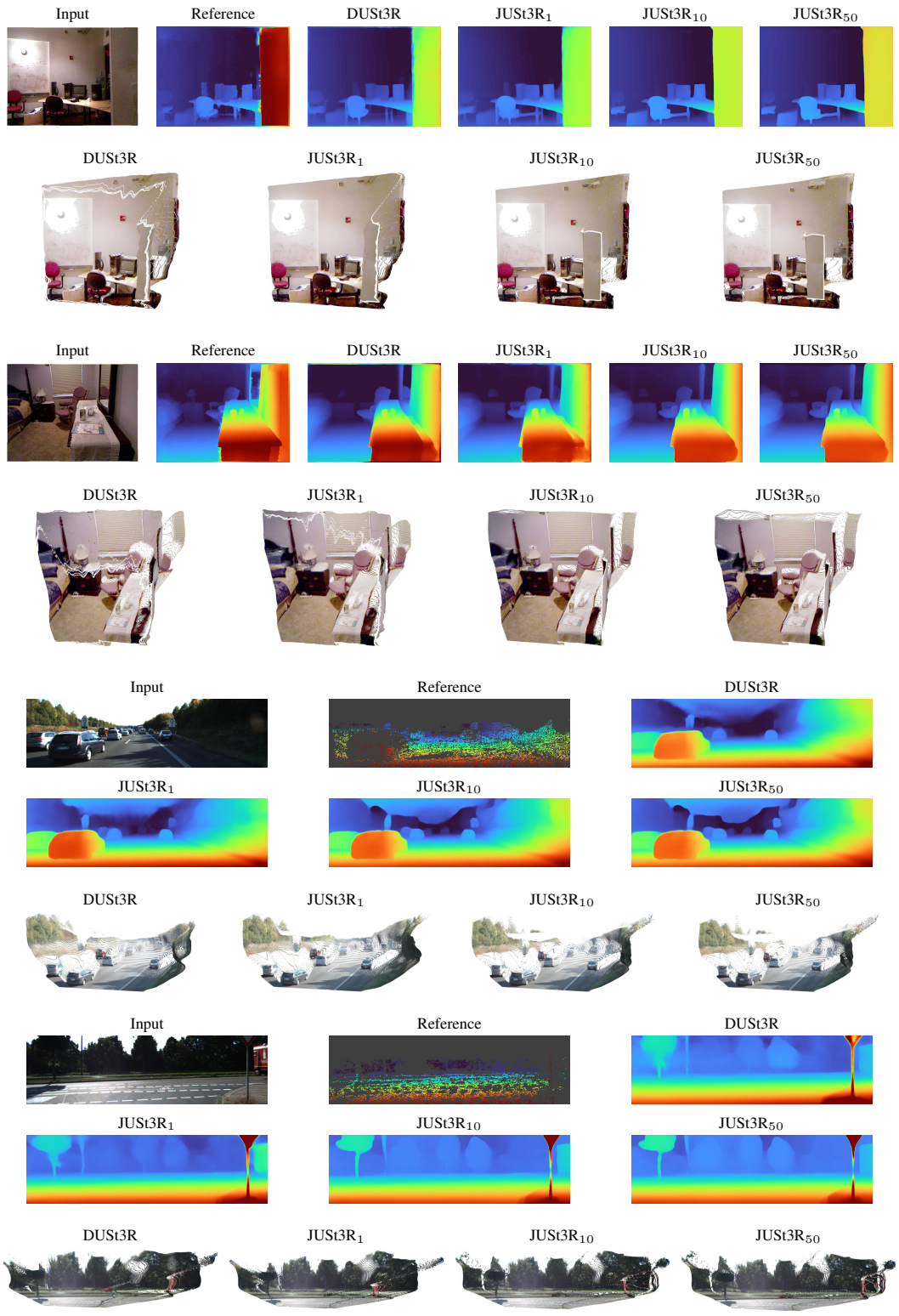

Figure 15: Additional JUSt3R comparisons, page 2. Each case shows depth predictions followed immediately by its offset point-cloud views.

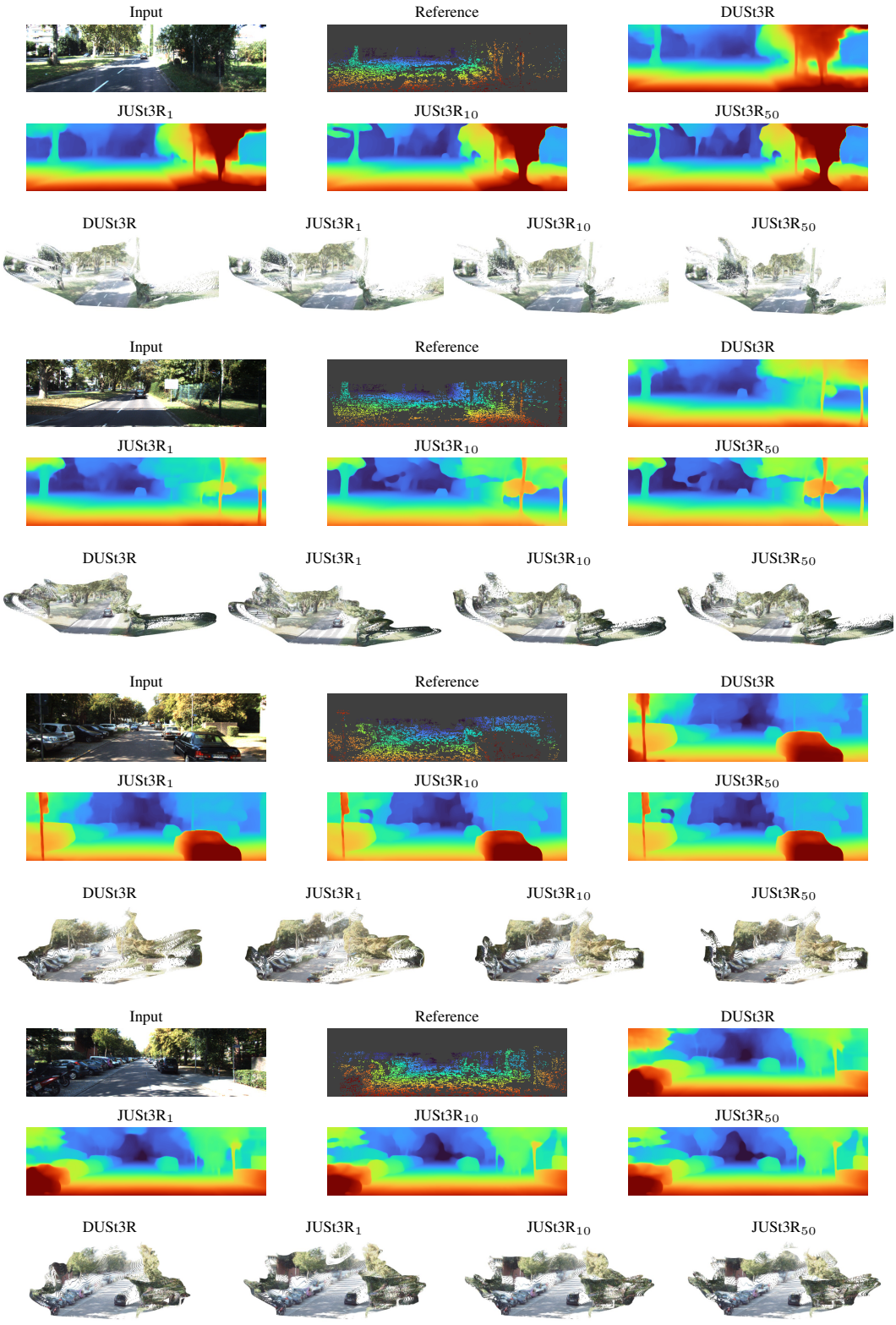

Figure 16: Additional JUSt3R comparisons, page 3. Each case shows depth predictions followed immediately by its offset point-cloud views.

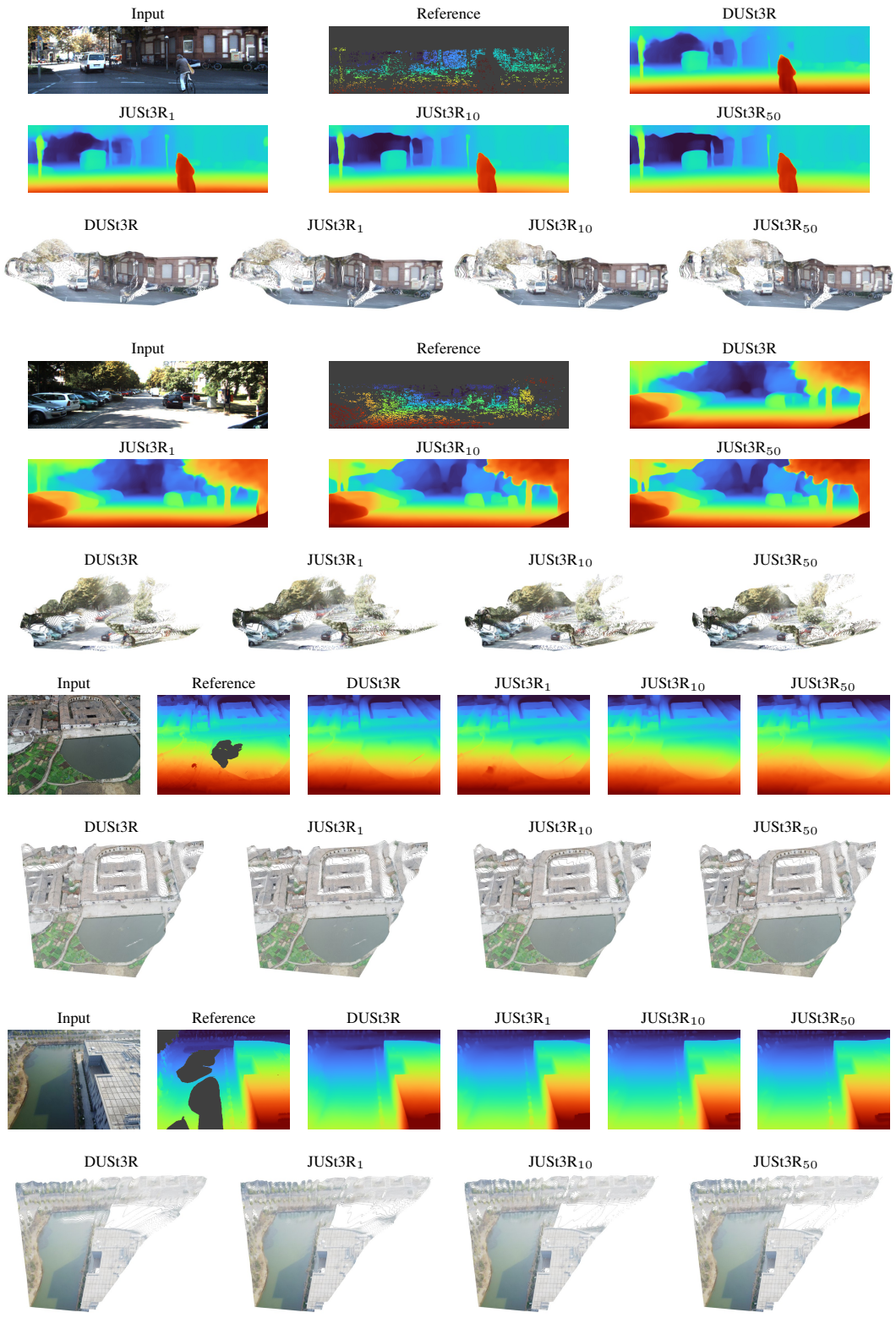

Figure 17: Additional JUSt3R comparisons, page 4. Each case shows depth predictions followed immediately by its offset point-cloud views.

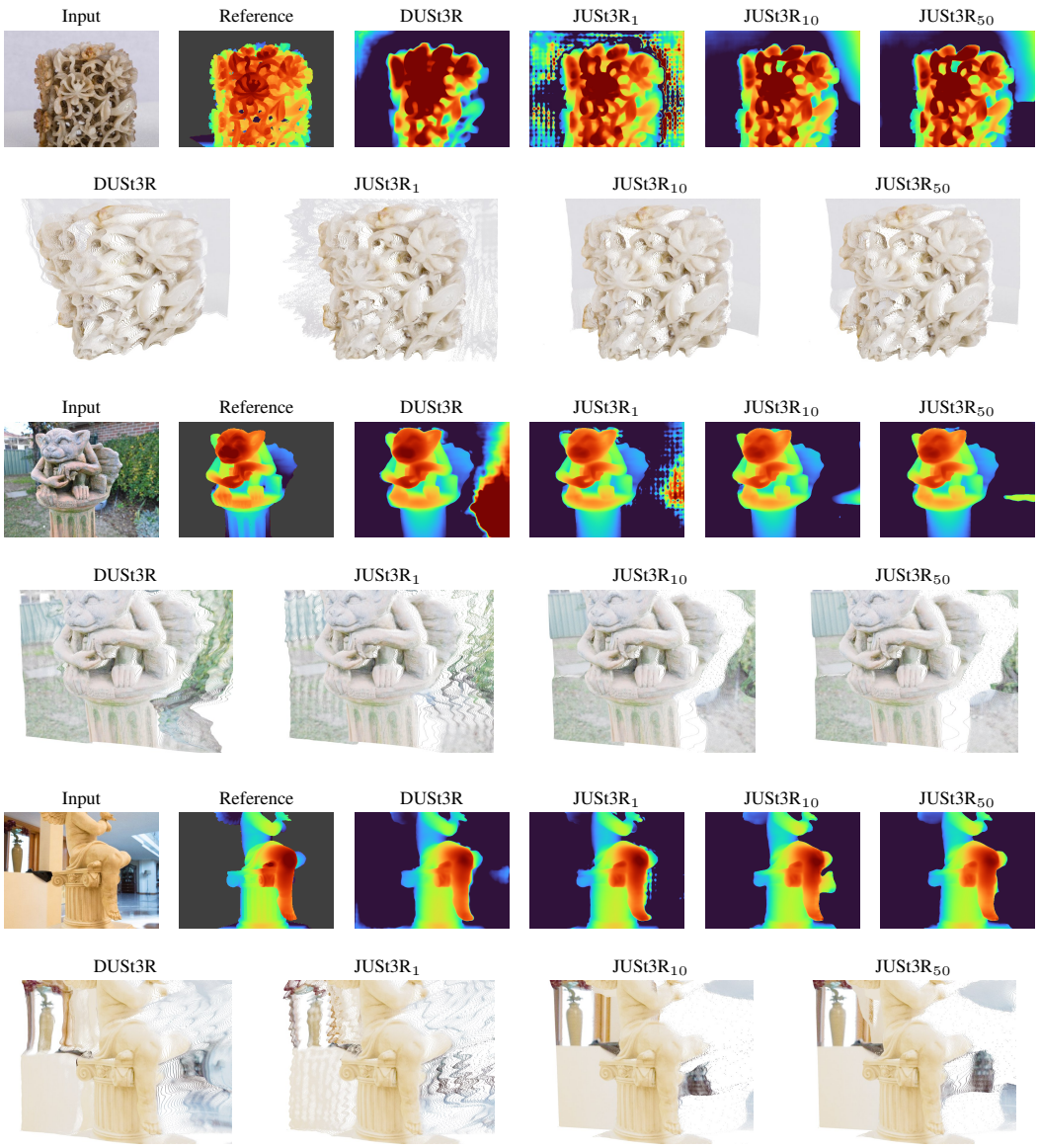

Figure 18: Additional JUSt3R comparisons, page 5. Each case shows depth predictions followed immediately by its offset point-cloud views.