REPRESENTATION-GUIDED STATE EVOLUTION FOR IMAGE-CONDITIONED POINT CLOUD RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-view 3D reconstruction seeks to recover complete object geometry from a single RGB image, yet remains inherently ambiguous because the observation reveals only a partial, viewpoint-dependent surface. Diffusion models offer a flexible generative formulation for this task, but point-cloud denoising requires a coherent global shape hypothesis that can be selectively updated with local image evidence. To address this challenge, we introduce RepStateDiff, a representation-guided conditional diffusion framework that formulates reconstruction as state-evolution denoising. By separating persistent shape retention from evidence-driven state updates, RepStateDiff preserves global structure while refining local geometry with image-aligned cues. It aligns frozen visual features with noisy 3D points and propagates hierarchical tokens through StateEvo, a linear-time sequence mixer; intermediate representation alignment and internal guidance further couple visual semantics to denoising. On ShapeNet-R2N2 and Pix3D, RepStateDiff achieves the strongest F1-scores across most evaluated settings and remains robust even with merely 1% of training data. On the three primary ShapeNet-R2N2 categories at 10% training data, it improves mean CD/F1 over PDM from 66.09/0.525 to 56.34/0.582, while PDM retains faster inference. Project page: `https://anonymous123-cyber.github.io/RepStateDiff`.

1 INTRODUCTION

Recovering 3D geometry from a single RGB image remains a long-standing inverse problem with applications in object digitization Sun et al. (2018), robotic perception Nie et al. (2020), augmented reality Weng et al. (2022), and interactive 3D content creation Tang et al. (2023). The problem is fundamentally under-constrained: a single view reveals only a partial surface, so unseen back surfaces, thin structures, concavities, and category-level regularities must be inferred from learned priors. Early single-view reconstruction methods have explored voxel Choy et al. (2016), mesh Wang et al. (2018), implicit Mescheder et al. (2019), and point-based representations Fan et al. (2017), each balancing geometric detail, topological flexibility, and computational efficiency. Building on these representations, recent image-to-3D systems emphasize efficient object reconstruction and generalizable visual-to-geometric alignment Cui et al. (2024), underscoring the practical importance of this task.

Point clouds provide a natural output representation for this setting because they describe surface samples directly without imposing a cubic voxel grid or a fixed mesh template. Generative point-cloud models, particularly diffusion-based approaches, are compelling because they capture complex shape distributions through progressive denoising Ho et al. (2020); Luo & Hu (2021); Zhou et al. (2021); Zeng et al. (2022); Nichol et al. (2022). This probabilistic formulation naturally accommodates single-view reconstruction, where multiple complete shapes may be compatible with the same limited observation. Accordingly, a broad body of recent work has extended diffusion models to point-cloud generation and completion Luo & Hu (2021); Zhou et al. (2021); Zeng et al. (2022); Nichol et al. (2022), projection-conditioned reconstruction Melas-Kyriazi et al. (2023), consistent conditioning Di et al. (2023), Bayesian diffusion variants Xu et al. (2024), semantic-cue mining Li et al. (2025a), recurrent diffusion Zhou et al. (2025), transformer-based RGB-to-point reconstruction Lee & Benes (2025), and part-aware latent diffusion Zhu et al. (2025); Zhou et al.

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.

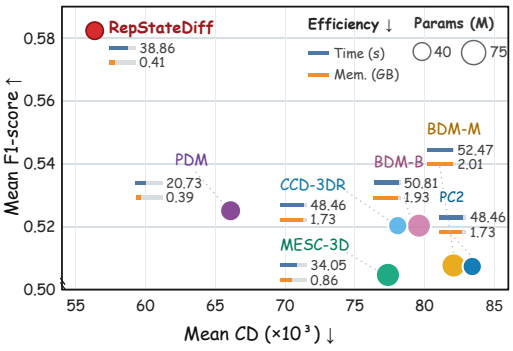

Figure 1: Absolute quality–efficiency comparison on ShapeNet-R2N2 with 10% training data. Marker positions indicate mean CD and F1-score, marker area denotes model parameters, and adjacent bars show sampling time and memory usage. RepStateDiff is highlighted in red.

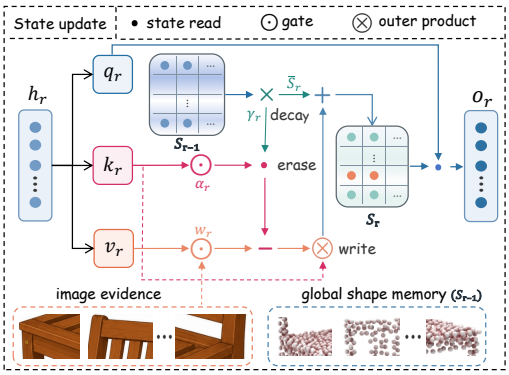

Figure 2: State update and readout of StateEvo. The preceding global shape memory is first decayed, then modified through key-aligned erasure and image-conditioned value writing before being queried for the token output.

(2026a). Point Diffusion Mamba (PDM) further combines diffusion with Mamba-based state-space modeling for data-scarce reconstruction, using local geometric aggregation, hierarchical feature integration, and dynamic weighted sampling to couple local details with generative priors Zhou et al. (2026b).

Despite this progress, diffusion-based single-view reconstruction involves two coupled challenges. First, image evidence and 3D shape priors play distinct roles during denoising. Visible cues provide local evidence for revising the evolving point hypothesis, whereas unobserved geometry requires a stable global shape memory. Conflating these signals into a single point-wise feature may obscure their distinct roles, particularly under limited supervision. Second, the denoising backbone must propagate object-level information across numerous point tokens at every diffusion step. Although accelerated samplers and consistency-style models reduce the number of reverse steps Song et al. (2021; 2023), the per-step denoiser still determines how efficiently geometry is propagated. Dense self-attention provides global interactions but incurs quadratic computational cost, whereas local graph operations are efficient but often miss long-range structural dependencies. Efficient sequence mixers, including linear attention Katharopoulos et al. (2020) and state-space point-cloud models Gu & Dao (2024); Liang et al. (2024); Mo (2025), offer a promising alternative. However, unlike ordinary sequences, point clouds introduce a complication: after serialization, adjacent tokens may correspond to spatially distant 3D regions.

These challenges call for a denoising architecture that decouples persistent information from dynamic evidence while allowing strong visual representations to influence the denoiser beyond its input layer. To this end, we propose RepStateDiff, a representation-guided conditional point-cloud diffusion framework that formulates single-view reconstruction as state-evolution denoising. Its central insight is to separate persistent shape retention from evidence-driven state updates, allowing image-aligned observations to refine local geometry without overwriting essential global shape memory. Specifically, RepStateDiff constructs hierarchical, surface-aligned point tokens from noisy coordinates and frozen image features Oquab et al. (2024). It then processes these tokens with StateEvo, a linear-time recurrent backbone whose channel-wise erase, write, and decay operations explicitly parameterize state removal, evidence injection, and memory retention.

The design also accounts for the unordered geometry of point sets and the temporal structure of diffusion sampling. Because serialized point tokens can abruptly shift between disjoint surfaces, a fixed recurrent memory schedule risks propagating information across regions that should remain weakly coupled. RepStateDiff therefore modulates memory decay using the 3D scan distance and diffusion timestep, while StateEvo preserves the separation between retained shape memory and local evidence updates. On the representation side, intermediate supervision and internal guidance allow image-aligned cues to shape both input conditioning and internal point-feature evolution. Figure 1 summarizes the trade-off: with 10% data, RepStateDiff improves mean CD/F1 over PDM on

the three primary ShapeNet-R2N2 categories and uses fewer parameters, while PDM retains faster inference and slightly lower memory use.

Our contributions are summarized as follows:

- We introduce StateEvo, a recurrent denoising backbone with explicit channel-wise erase, write, and decay gates that decouple persistent shape retention from image-conditioned evidence updates in point-cloud diffusion.
- We develop RepStateDiff, a conditional reconstruction framework that aligns frozen visual features with noisy 3D points, constructs hierarchical point tokens, and guides their evolution throughout denoising.
- We demonstrate that representation-guided state evolution consistently improves reconstruction quality while maintaining a compact memory footprint on ShapeNet-R2N2 and Pix3D, particularly in extreme low-data regimes.

2 Related Work

2.1 Single-view 3D Reconstruction

Single-view 3D reconstruction spans voxel, mesh, implicit, and point-cloud representations. While voxel methods support straightforward 3D convolutions, they are bottlenecked by cubic memory growth Choy et al. (2016); Xie et al. (2019). Conversely, mesh and surface-patch methods recover explicit surfaces but offer limited topological flexibility Wang et al. (2018); Groueix et al. (2018). Implicit representations improve surface continuity yet require an extraction stage before evaluation Mescheder et al. (2019). In contrast, point-set prediction avoids fixed grids and rigid templates by directly synthesizing surface samples Chang et al. (2015); Fan et al. (2017). To bridge the visual–geometric gap, recent systems have incorporated stronger visual priors Liu et al. (2023); Long et al. (2024), image–geometry alignment Cui et al. (2024), semantic cues Li et al. (2025a), deterministic RGB-to-point Transformers Lee & Benes (2025), and object- or scene-aware modeling Huang et al. (2026); Wang et al. (2026). Other methods target alternative output representations: SPAR3D uses a lightweight point-diffusion stage as an intermediate proxy for mesh reconstruction Huang et al. (2025), whereas GSD diffuses sets of 3D Gaussian primitives for view-guided reconstruction Mu et al. (2024). Ultimately, because a single view captures only partial geometry, all such approaches face the shared challenge of integrating local image evidence with expressive shape priors.

2.2 Diffusion-based Point-cloud Reconstruction

Diffusion-based point-cloud reconstruction formulates single-view recovery as conditional denoising in 3D coordinate space, progressively transforming noisy points into an image-consistent shape Ho et al. (2020); Melas-Kyriazi et al. (2023). Existing work has mainly improved how visual observations guide this reverse process through repeated 2D–3D feature alignment, consistency-aware conditioning, uncertainty-aware formulations, and transformer- or recurrence-based denoisers Melas-Kyriazi et al. (2023); Di et al. (2023); Xu et al. (2024); Feng et al. (2024); Zhou et al. (2025). This line of research primarily emphasizes conditioning design and denoiser architecture, leaving persistent shape memory and evidence-driven revision implicit. RepStateDiff instead targets this internal state evolution by explicitly separating retention of the global shape hypothesis from image-guided local updates.

2.3 Efficient Sequence Modeling

Efficient sequence modeling is crucial because the denoising backbone must iteratively process large sets of point tokens. To avoid the quadratic cost of dense attention, linear attention and selective state-space models use recurrent updates Katharopoulos et al. (2020); Gu & Dao (2024). Specifically, linear attention accumulates key–value statistics, Mamba controls discretized state-space transitions through input-selective parameters, and RWKV uses time-mixed key–value aggregation with a receptance-gated readout Katharopoulos et al. (2020); Gu & Dao (2024); Peng et al. (2023). PDM is a recent direct baseline for data-efficient single-view reconstruction: it combines a dual-directional state-space module with Local Geometric Aggregation, a Hierarchical Feature Integration Network,

and Dynamic Weighted Sampling Zhou et al. (2026b). RepStateDiff instead makes state retention and evidence-driven updates explicit through separate erase, write, and decay gates, with retention modulated by 3D token displacement and diffusion time. While recent linear-time sequence models offer efficient state propagation, applying generic 1D sequence gating directly to unordered 3D point clouds remains challenging because token serialization can introduce severe spatial discontinuities. StateEvo addresses this issue by decoupling key-directed erasure from value-side writing and modulating channel-wise retention according to spatial displacement and diffusion time. While PointNet++ captures local geometry through hierarchical grouping Qi et al. (2017b), recent state-space and RWKV adaptations show that sequence-style mixing is viable for 3D tasks Liang et al. (2024); Mo (2025); He et al. (2025). RepStateDiff advances these approaches by adding retention modulation and image-conditioned state updates, ensuring that recurrent propagation accounts for both serialized order and local geometric evidence.

3 METHOD

RepStateDiff reconstructs 3D point clouds from single RGB images using coordinate-space conditional diffusion. As shown in Figure 3, each reverse step is formulated as a state-evolution process: the denoiser retains a coherent global shape hypothesis while selectively integrating image-aligned local evidence. At the token level, StateEvo decays the preceding global shape memory, erases obsolete context, and writes image-conditioned evidence into the recurrent state, as detailed in Figure 2. The remainder of this section presents the designs of surface-aligned conditioning, the StateEvo mixer, and representation-guided denoising.

3.1 SURFACE-ALIGNED HIERARCHICAL POINT CONDITIONING

A central challenge in single-view reconstruction is that image evidence and 3D state variables play fundamentally distinct roles during denoising. Specifically, image features provide local visible-surface cues and category-level priors, whereas point coordinates define the evolving spatial hypothesis that must remain coherent across the object. To address this disparity, RepStateDiff constructs a surface-aligned hierarchical point representation in which frozen visual features are explicitly projected onto noisy points before recurrent state propagation.

Let Φ denote a frozen image encoder Oquab et al. (2024). We first extract the dense image feature map $F_I = \Phi(I)$ by aggregating the final K normalized intermediate layers, thereby combining local patch details with higher-level visual semantics. For each noisy point $x_i^t \in X_t$, camera-based surface rasterization provides the local image condition:

$$c_i = \mathrm{RasterProj}_C(F_I, x_i^t) \tag{1}$$

The operator $\mathrm{RasterProj}_C$ projects x_i^t under camera parameters C and retrieves the corresponding feature for each visible point. Unlike a static nearest-pixel lookup on a precomputed depth map, this rasterization procedure ties visual conditioning to the current noisy point hypothesis. Points that move during denoising therefore receive updated image evidence. This mechanism is crucial because the network iteratively revises the same shape rather than decoding a point cloud in a single pass. Finally, the denoiser input is formed by concatenating x_i^t, the projected RGB values, and c_i, providing each point with its current 3D hypothesis and aligned 2D observation. Rasterization and point-wise feature-assignment details are provided in Appendix Sec. F.1.

Multi-scale tokenization uses farthest point sampling (FPS) and K-nearest-neighbor (KNN) grouping. At scale s, let p_j^s denote the j-th sampled center and $\mathcal{G}_j^s = \{x_{j,k}^t - p_j^s,\ c_{j,k}\}_{k=1}^K$ represent its centered local group, including both relative coordinates and associated image features. Each group is encoded as

$$h_j^s = \mathrm{PatchEnc}\left(\mathcal{G}_j^s\right) + \phi_{pos}(p_j^s) + \phi_t(t) \tag{2}$$

where $\mathrm{PatchEnc}$ is a Mini-PointNet Qi et al. (2017a), ϕ_{pos} is an MLP-based positional encoding, and ϕ_t is a sinusoidal timestep embedding followed by an MLP. This design equips each token with local geometry, pixel-aligned visual evidence, spatial position, and diffusion-time information before global denoising. The conditioning pathway remains single-stream: these cues are fused early into hierarchical tokens, while StateEvo handles long-range interactions and separates state retention from evidence writing within its recurrent updates.

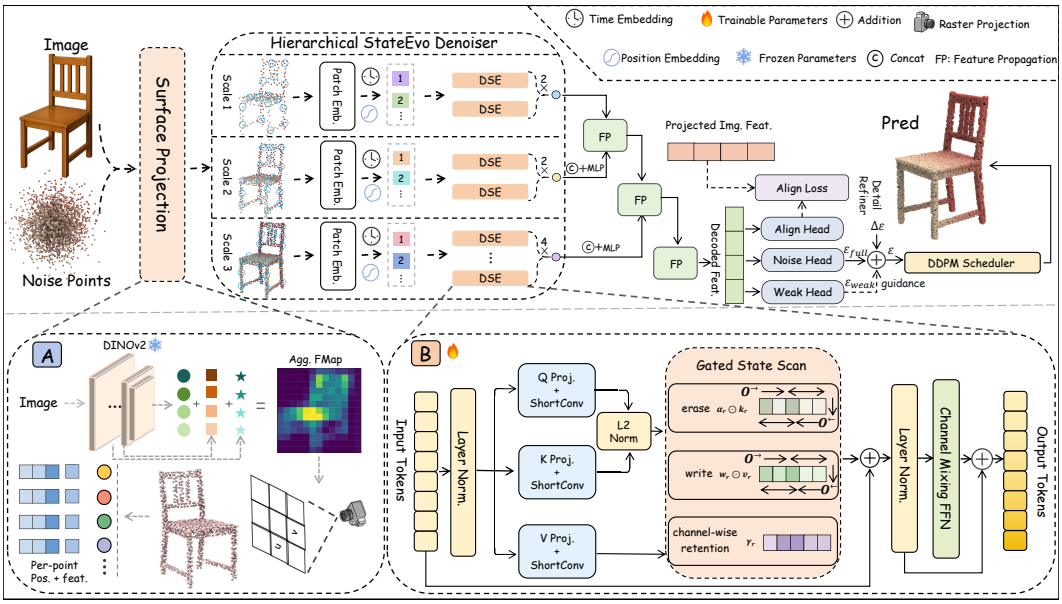

Figure 3: Architectural overview of RepStateDiff. Given an RGB image and noisy 3D points, surface projection aligns frozen image features with the current point hypothesis, associating each point with local appearance and semantic context. Hierarchical tokenization then organizes these image-conditioned points into multi-scale tokens for recurrent denoising. StateEvo propagates the global shape state while selectively updating it with local visual evidence. Finally, the representation-guided heads and a local refiner transform the updated point features into the noise prediction for the next denoising step.

3.2 State Evolution for Point-Token Denoising

After hierarchical tokenization, the denoiser must propagate geometric information across serialized point tokens. While full self-attention provides global connectivity, it scales quadratically with the number of tokens; conversely, local graph operations are efficient but often miss long-range shape dependencies. RepStateDiff addresses these limitations with StateEvo, a linear-time recurrent spatial mixer that maintains a persistent geometric state and updates it through explicit erase, write, and decay operations. Rather than serving as generic sequence memory, this recurrent state retains a coherent global shape hypothesis while accommodating evidence-driven local edits.

Let r index the token order, with t denoting the diffusion timestep. Given the input sequence $H = [h_1, \ldots, h_L]$, StateEvo constructs activated short-convolutional projections and normalizes per-head features as follows:

$$\begin{cases} z_r^u = \mathrm{SiLU}(\mathcal{C}_u(W_u H)_r), \quad u \in \{q, k, v\}, \\ q_r = D_k^{-1/2} \dfrac{z_r^q}{\|z_r^q\|_2}, \quad k_r = \dfrac{z_r^k}{\|z_r^k\|_2}, \quad v_r = z_r^v \end{cases} \tag{3}$$

The state update uses three channel-wise gates: the erase gate α_r, the write gate w_r, and the decay gate γ_r:

$$\begin{cases} \alpha_r = \sigma(W_\alpha h_r), \\ w_r = \sigma(W_w h_r), \\ g_r = -\log\left(1 + \exp\left(W_g h_r + \delta\right)\right) \odot \exp(a), \\ \gamma_r = \exp(g_r) \end{cases} \tag{4}$$

Here, $q_r, k_r, \alpha_r, \gamma_r \in \mathbb{R}^{D_k}$ and $v_r, w_r \in \mathbb{R}^{D_v}$, while $S_r \in \mathbb{R}^{D_k \times D_v}$ denotes the recurrent state matrix. The learned vector a parameterizes the channel-wise decay rates, and δ is the learned bias of the decay activation. The erase gate α_r determines which previous state content is removed along the key axis, whereas the write gate w_r controls which new value content is integrated. The log-decay $g_r \leq 0$ defines the memory retention factor $\gamma_r \in (0, 1]$. The previous state is first decayed:

$$\bar{S}_r = \mathrm{diag}(\gamma_r) S_{r-1} \tag{5}$$

Table 1: ShapeNet-R2N2 results on three primary categories. CD is multiplied by 10^3; lower CD and higher F1 are better.

Method	Chair								Airplane								Car							
	1%		10%		50%		100%		1%		10%		50%		100%		1%		10%		50%		100%	
	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
PC2 Melas-Kyriazi et al. (2023, *CVPR*)	108.61	0.381	97.25	0.393	73.58	0.437	65.57	0.464	117.09	0.579	88.00	0.605	76.39	0.628	65.97	0.655	66.34	0.509	64.99	0.524	62.59	0.542	64.36	0.547
CCD-3DR Di et al. (2023, *arXiv*)	–	–	89.79	0.418	63.13	0.474	58.47	0.498	–	–	81.29	0.612	72.46	0.635	62.77	0.651	–	–	63.13	0.531	62.25	0.550	61.88	0.562
BDM-M Xu et al. (2024, *CVPR*)	107.66	0.382	94.94	0.395	71.56	0.446	64.48	0.468	117.42	0.580	87.75	0.604	73.19	0.629	65.16	0.653	66.56	0.509	63.53	0.524	60.71	0.549	64.16	0.554
BDM-B Xu et al. (2024, *CVPR*)	107.64	0.3914	94.67	0.410	69.99	0.463	64.21	0.485	115.10	0.5878	83.62	0.612	68.66	0.641	59.04	0.660	65.27	0.5188	60.48	0.539	62.58	0.554	65.85	0.559
MESC-3D Li et al. (2025a, *CVPR*)	192.32	0.2444	101.51	0.381	73.47	0.427	65.69	0.458	**87.77**	0.5013	74.31	0.611	51.28	0.619	50.54	0.714	70.32	0.4347	56.28	0.522	**44.48**	0.532	51.99	0.602
PDM Zhou et al. (2026b, *ECCV*)	115.43	0.3702	82.41	0.419	68.85	0.465	62.14	0.488	92.63	0.5619	60.64	0.614	50.14	0.663	48.66	0.719	**61.58**	0.5084	55.23	0.542	55.53	0.558	51.54	0.607
RepStateDiff (Ours)	**97.76**	**0.405**	**64.96**	**0.481**	**50.50**	**0.525**	**46.38**	**0.541**	92.72	**0.599**	**50.66**	**0.691**	**40.54**	**0.732**	**37.55**	**0.743**	62.53	**0.519**	**53.42**	**0.575**	50.84	**0.610**	**50.11**	**0.618**

The decayed state is then updated through coordinated erase and write operations:

$$S_r = \bar{S}_r + k_r \left(w_r \odot v_r - \bar{S}_r^\top (\alpha_r \odot k_r) \right)^\top \tag{6}$$

Let $\mathcal{R}_\rightarrow$ and $\mathcal{R}_\leftarrow$ denote the recurrence operators applied in the forward and reverse token orders, respectively, with reversed outputs aligned to the original order, and let $\mathcal{G} = (\boldsymbol{\alpha}, \boldsymbol{w}, \boldsymbol{g})$. The token-level and chained sequence readouts are computed as:

$$\begin{cases} o_r = S_r^\top q_r, \quad O^\rightarrow = \mathcal{R}_\rightarrow(Q, K, V; \mathcal{G}), \\ O^\leftarrow = \mathcal{R}_\leftarrow(Q, K, O^\rightarrow; \mathcal{G}), \quad O = \frac{1}{2}(O^\rightarrow + O^\leftarrow) \end{cases} \tag{7}$$

This recurrence propagates long-range geometric context in linear time by accumulating mappings from key-aligned patterns to values, enabling object-level retrieval without pairwise attention. To complement the forward context, the backward scan uses $O^\rightarrow$ as its value sequence and refines it in reverse order; averaging the two retains bidirectional context. In each scan, the erase operation attenuates key-conflicting state content, whereas the write operation incorporates image- and point-conditioned evidence, keeping memory retention distinct from state editing.

Point-cloud tokens lack a natural sequential order: after FPS and serialization, consecutive tokens may correspond to spatially distant 3D regions. To address this issue, RepStateDiff modulates memory decay according to 3D spatial distance and the diffusion timestep.

Channel-wise decay rates are initialized as a log-spaced spectrum, providing diverse memory scales at the start of training:

$$A_d = \exp\left(\log A_{\min} + \frac{d}{D_h - 1} (\log A_{\max} - \log A_{\min}) \right) \tag{8}$$

Here, D_h denotes the per-head channel dimension, $d \in \{0, \ldots, D_h - 1\}$ indexes channels within each head, and $A_{\min}$ and $A_{\max}$ define the lower and upper bounds of the decay spectrum. The base log-decay is then dynamically adapted by:

$$\begin{cases} g'_r = \rho(t) \left(g_r - |\boldsymbol{\lambda}_{\text{dist}}| \, \|p_r - p_{r-1}\|_2 \right), \\ \rho(t) = \exp(-\lambda_{\text{time}} t / T) \end{cases} \tag{9}$$

where p_r is the 3D center of token r, $\boldsymbol{\lambda}_{\text{dist}} \in \mathbb{R}^{D_k}$ is a learned non-negative distance coefficient applied channel-wise, and λ_{time} is a learned scalar controlling diffusion-time modulation. The adapted g'_r replaces g_r when computing γ_r, explicitly reducing state retention across large spatial shifts and adjusting memory persistence along the denoising trajectory. The log-spaced initialization lets different channels capture distinct default memory horizons, while the spatial and temporal terms adapt these horizons to the current point ordering and diffusion step. This design keeps the recurrence lightweight: no additional attention map is constructed, and the modulation only regulates how much historical state is carried forward.

3.3 Representation-Guided Denoising and Detail Refinement

Surface projection provides point-wise visual conditioning at the input layer; however, the recurrent backbone may still develop internal representations that drift from the frozen visual features. To address this, RepStateDiff introduces two auxiliary objectives that explicitly couple internal point features to the visual representation space.

First, intermediate representation alignment encourages the decoded point features to remain consistent with the frozen visual representation. Let $Z_\theta(X_t, I, C, t) \in \mathbb{R}^{N \times D_z}$ denote the decoded point

features, let D_{img} denote the image-feature dimension, and let $\psi : \mathbb{R}^{D_z} \to \mathbb{R}^{D_{\text{img}}}$ be a learned projection. With $V_I = \text{RasterProj}_C(F_I, X_t) \in \mathbb{R}^{N \times D_{\text{img}}}$, we define the alignment loss as:

$$\mathcal{L}_{\text{align}} = \|\text{norm}\left(\psi(Z_\theta)\right) - \text{norm}\left(V_I\right)\|_2^2 \tag{10}$$

This loss leaves the frozen image encoder unchanged; instead, it guides internal point features toward the representation manifold used for conditioning. The objective is applied to decoded point features rather than raw coordinates, thereby regularizing the denoiser's internal representation without replacing the main diffusion loss. It is particularly effective because projected image features carry both local and semantic information, allowing the alignment target to constrain visible-surface detail and category-level context simultaneously.

Second, an internal guidance mechanism augments the primary noise-prediction network with a lightweight auxiliary head. Let g_θ be a compact MLP attached to the decoded point features, producing $\epsilon_{\text{weak}} = g_\theta(Z_\theta)$. During training, g_θ is supervised by an auxiliary denoising objective, yielding the overall loss:

$$\mathcal{L} = \mathcal{L}_{\text{diff}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{weak}} \left\|\epsilon - \epsilon_{\text{weak}}\right\|_2^2 \tag{11}$$

During sampling, this auxiliary prediction provides a representation-dependent guidance direction:

$$\hat{\epsilon}_{\text{guided}} = \hat{\epsilon} + s_{\text{ig}} \left(\hat{\epsilon} - \epsilon_{\text{weak}}\right) \tag{12}$$

Here, s_{ig} controls the guidance scale. This mechanism encourages Z_θ to support direct noise prediction independently, thereby coupling the internal representation tightly to the denoising objective. Because the auxiliary head supplies a correction direction derived from the same point features as the main denoiser, the guidance operates entirely internally. This approach requires neither an external classifier nor an additional image encoder at inference.

Finally, a multi-scale local detail refiner computes a residual correction from the surface-projected image features:

$$\hat{\epsilon} = f_\theta(X_t, I, C, t) + R_\theta(X_t, I, C, t) \tag{13}$$

Here, R_θ samples a hierarchical pyramid of image features at the projected point locations and fuses them with the 3D point coordinates, depth values, and timestep embeddings. The refiner is initialized to output a zero residual, allowing it to enhance visible-surface details progressively without destabilizing early training.

4 Experiments

To evaluate reconstruction quality, low-data behavior, component contributions, and efficiency, we follow the BDM protocol Xu et al. (2024) on synthetic ShapeNet Chang et al. (2015); Choy et al. (2016) and real-image Pix3D Sun et al. (2018), while including the recent PDM method as a direct published baseline. Ground-truth shapes contain 4,096 points; images are resized to 224×224; and we report CD and F1 at $\tau = 0.01$ following established protocols Melas-Kyriazi et al. (2023); Xu et al. (2024). PDM uses the same categories, 80/20 split, point count, resolution, and threshold, and its published results are included for comparison Zhou et al. (2026b).

All category-specific models are optimized with AdamW and trained on a single NVIDIA RTX 3090 GPU. To ensure strict reproducibility within the main-text page limits, comprehensive hyperparameter configurations—including exact loss weighting ($\lambda_{align}, \lambda_{weak}$), internal guidance scales (s_{ig}), DDPM noise schedules, multi-scale token allocations, and sampling steps—are documented in Appendix E. The complete training and inference codebase is also provided in the supplementary material. Due to space constraints, Appendix B presents the additional ShapeNet-R2N2 categories and joint training (Secs. B.2–B.3), the extended Pix3D results (Sec. B.4), resolution/NFE efficiency and recent diffusion comparisons (Secs. B.8–B.9), feature-depth and StateEvo controls for hierarchy, serialization, retention, scan direction, and backbone (Secs. B.10–B.12), and exact EMD (Sec. B.13). Beyond these two main benchmarks, we also evaluate CO3Dv2 and Objaverse: the CO3Dv2 data-scaling and 13-category full-data results are reported in Appendices B.5–B.6 (Table 15), while the 12-category Objaverse results are reported in Appendix B.7 (Table 16).

Table 2: ShapeNet-R2N2 10% results on the remaining ten categories. CD is multiplied by 10^3; lower CD and higher F1 are better.

Method	Bench		Cabinet		Display		Lamp		Loudspeaker	
	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
BDM-B Xu et al. (2024, *CVPR*)	80.93	0.5420	97.98	0.3410	119.32	0.5190	211.54	0.3410	118.87	0.3330
MESC-3D Li et al. (2025a, *CVPR*)	180.38	0.3370	203.14	0.2500	175.57	0.3810	212.86	0.2320	240.13	0.2470
RepStateDiff (Ours)	**60.23**	**0.5990**	**59.55**	**0.4320**	**77.71**	**0.5380**	**137.40**	**0.4350**	**92.11**	**0.3730**

Method	Rifle		Sofa		Table		Telephone		Watercraft	
	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
BDM-B Xu et al. (2024, *CVPR*)	39.76	0.7560	65.39	0.4380	126.40	0.4330	62.98	**0.6330**	58.35	0.5540
MESC-3D Li et al. (2025a, *CVPR*)	92.80	0.7270	166.62	0.3290	239.83	0.2170	144.11	0.4900	149.00	0.4230
RepStateDiff (Ours)	**37.83**	**0.7730**	**52.16**	**0.4770**	**67.50**	**0.5580**	**50.62**	0.6270	**48.23**	**0.5810**

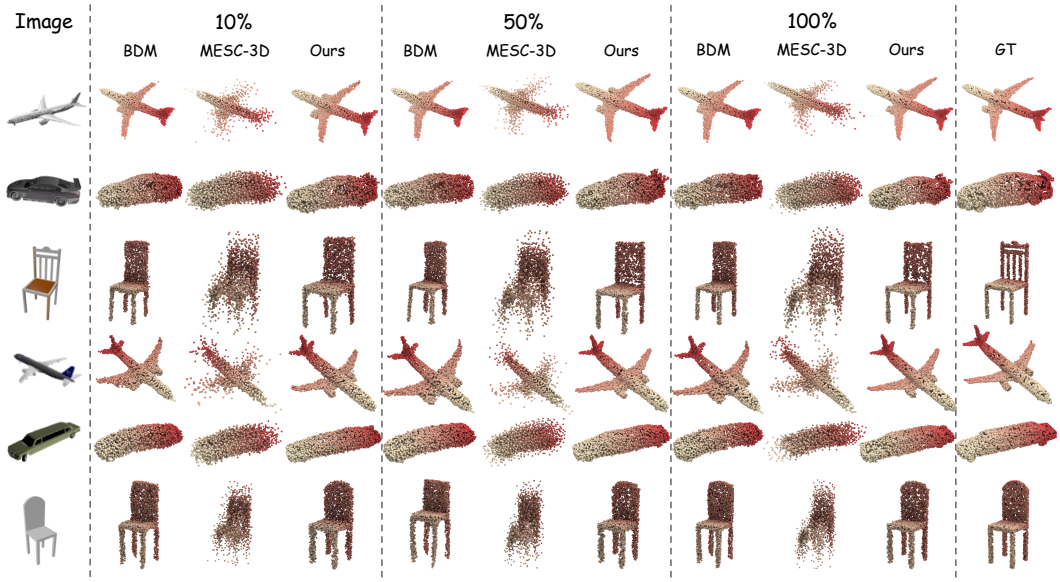

Figure 4: Qualitative ShapeNet-R2N2 comparison across 10%, 50%, and 100% training-data regimes using the available BDM and MESC-3D outputs. PDM is included in the quantitative comparison in Table 1.

4.1 Quantitative Results

Table 1 compares RepStateDiff with PDM and other published baselines across training-data budgets. At 10%, RepStateDiff achieves the lowest CD and highest F1 in all three primary categories; averaged over these categories, CD/F1 improves from 66.09/0.525 for PDM to 56.34/0.582. From 10% to 100% data, it outperforms PDM on both metrics in every category. At 1%, it improves F1 in all three categories and CD for *chair*, while PDM has lower CD for *airplane* and *car*. At 50%, RepStateDiff has the highest F1 in each category and lower CD than PDM in all three; among all listed methods, MESC-3D retains lower *car* CD (44.48 versus 50.84). At 100%, RepStateDiff leads all listed methods on both metrics across the three categories.

Table 2 extends the 10% evaluation to the remaining ten ShapeNet-R2N2 categories. RepStateDiff obtains the best CD in all ten categories and the best F1 in nine; the only exception is Telephone, where BDM-B has a slightly higher F1 (0.633 versus 0.627). This consistent CD improvement across both compact and thin-structure objects indicates that the state-evolution design generalizes beyond the three primary categories.

Table 3 compares the methods on real images. Relative to PDM, RepStateDiff lowers chair CD from 79.28 to 57.83 and table CD from 184.35 to 153.77, while obtaining higher F1 in all three categories; PDM has slightly lower sofa CD (41.43 versus 41.72). The seven additional Pix3D categories are evaluated against the available BDM-B and MESC-3D results in Appendix B.4; comparable PDM results are unavailable for those categories.

Table 3: Pix3D reconstruction results. CD is multiplied by 10^3; lower CD and higher F1 are better. See Appendix Tables 13 for additional 10% categories.

Method	Chair		Sofa		Table	
	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
PC2 Melas-Kyriazi et al. (2023, *CVPR*)	115.94	0.443	47.17	0.445	202.77	0.397
CCD-3DR Di et al. (2023, *arXiv*)	111.42	0.456	44.91	0.450	196.28	0.418
BDM-M Xu et al. (2024, *CVPR*)	113.40	0.449	44.50	0.451	202.08	0.413
BDM-B Xu et al. (2024, *CVPR*)	110.60	0.455	45.05	0.455	186.46	0.429
MESC-3D Li et al. (2025a, *CVPR*)	91.36	0.370	41.98	0.284	206.16	0.306
PDM Zhou et al. (2026b, *ECCV*)	79.28	0.499	**41.43**	0.463	184.35	0.422
RepStateDiff (Ours)	**57.83**	**0.596**	41.72	**0.469**	**153.77**	**0.436**

Table 4: Component ablation on 10% ShapeNet-R2N2 (mean over chair, airplane, and car). A checkmark indicates that the component is enabled. S, M, A, and G denote StateEvo, multi-layer visual features, intermediate alignment, and internal guidance, respectively. Additional ablations appear in Appendix Table 22 and Tables 24–30.

Config.	S	M	A	G	Chair CD↓	Chair F1↑	Mean CD↓	Mean F1↑
Base	–	–	–	–	80.01	0.4314	65.52	0.5448
S	✓	–	–	–	73.35	0.4528	61.94	0.5611
S+M	✓	✓	–	–	71.25	0.4582	59.06	0.5684
S+A	✓	–	✓	–	78.09	0.4483	63.76	0.5604
S+G	✓	–	–	✓	76.18	0.4538	63.49	0.5633
S+A+G	✓	–	✓	✓	76.07	0.4573	64.05	0.5638
S+M+G	✓	✓	–	✓	65.74	0.4785	56.45	0.5818
S+M+A	✓	✓	✓	–	69.67	0.4611	58.75	0.5676
Full	✓	✓	✓	✓	**64.96**	**0.4814**	**56.34**	**0.5824**

Table 5: Efficiency comparison (inference time per object and additional peak GPU memory). See Appendix Tables 17–19 for NFE, FLOPs, and resolution-dependent results.

Method	Params (M)↓	Time (s)↓	Memory (GB)↓
PC2 / CCD-3DR	47.41	48.46	1.73
BDM-M	74.82	52.47	2.01
BDM-B	73.78	50.81	1.93
MESC-3D	74.84	34.05	0.86
PDM	56.34	**20.73**	**0.39**
RepStateDiff	**38.11**	38.86	0.41

4.2 Qualitative Results

Figure 4 provides qualitative context for the available BDM and MESC-3D outputs, whereas PDM is compared quantitatively in Table 1.

4.3 Ablation and Efficiency

Table 4 reports component ablations under the 10% ShapeNet-R2N2 setting. Adding StateEvo lowers chair CD from 80.01 to 73.35 and raises F1 from 0.4314 to 0.4528; mean CD/F1 improve from 65.52/0.5448 to 61.94/0.5611. Appendix Sec. B.12 reports further serialization, distance-modulation, and decay ablations (Tables 26 and 28).

Representation guidance is conditional on multi-layer features: alignment and guidance alone can worsen CD, but with multi-layer features, guidance lowers mean CD from 59.06 to 56.45, and alignment yields the best full-model scores by exploiting complementary local and semantic cues.

Table 5 shows the trade-off against PDM: RepStateDiff uses fewer parameters (38.11M versus 56.34M), whereas PDM has lower reported sampling time (20.73s versus 38.86s) and slightly lower memory use (0.39GB versus 0.41GB). RepStateDiff's advantage is reconstruction quality under limited training data rather than inference speed or memory.

Across evaluations, F1 gains are strongest on the *car* and Pix3D *sofa*, while CD gains concentrate on the *chair* and *table*; compact StateEvo conditioning and representation guidance improve point features without a larger backbone, supporting persistent state refinement.

5 Conclusion

We presented RepStateDiff, a conditional point-cloud diffusion framework that separates persistent shape context from image-guided local updates. StateEvo retains context across serialized tokens and edits it with image-aligned evidence, while representation guidance couples this evolution to visual features. On the three primary ShapeNet-R2N2 categories at 10% data, RepStateDiff improves mean CD/F1 over PDM from 66.09/0.525 to 56.34/0.582 and outperforms it on both metrics from 10% to 100%; at 1%, it improves F1 in all three categories. On Pix3D, it improves F1 in all three main categories and CD on chair and table, while PDM remains faster and slightly more memory-efficient. Ablations support the complementary roles of StateEvo and representation guidance.

AI USE STATEMENT

Generative AI tools were used only to assist with LaTeX format typesetting and language-level proofreading. The authors remain responsible for all scientific claims, experimental results, citations, and final manuscript content, and verified the generated formatting and compilation output.

REFERENCES

Yuta Baba and Keiji Yanai. Point-MF: One-step point cloud generation from a single image via mean flows, 2026. URL https://arxiv.org/abs/2604.24586.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3d model repository, 2015. URL https://arxiv.org/abs/1512.03012.

Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision*, pp. 628–644, 2016. doi:10.1007/978-3-319-46484-8_38.

Ruikai Cui, Xibin Song, Weixuan Sun, Senbo Wang, Weizhe Liu, Shenzhou Chen, Taizhang Shang, Yang Li, Nick Barnes, Hongdong Li, and Pan Ji. LAM3D: Large image-point clouds alignment model for 3d reconstruction from single image. In *Advances in Neural Information Processing Systems*, pp. 4454–4480, 2024. doi:10.52202/079017-0145.

Yan Di, Chenyangguang Zhang, Pengyuan Wang, Guangyao Zhai, Ruida Zhang, Fabian Manhardt, Benjamin Busam, Xiangyang Ji, and Federico Tombari. CCD-3DR: Consistent conditioning in diffusion for single-image 3d reconstruction, 2023. URL https://arxiv.org/abs/2308.07837.

Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 605–613, 2017. doi:10.1109/CVPR.2017.264.

Yu Feng, Xing Shi, Mengli Cheng, and Yun Xiong. DiffPoint: Single and multi-view point cloud reconstruction with ViT based diffusion model, 2024. URL https://arxiv.org/abs/2402.11241.

Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. AtlasNet: A papier-mache approach to learning 3d surface generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 216–224, 2018. doi:10.1109/CVPR.2018.00030.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *COLM*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Qingdong He, Jiangning Zhang, Jinlong Peng, Haoyang He, Xiangtai Li, Yabiao Wang, and Chengjie Wang. PointRWKV: Efficient RWKV-like model for hierarchical point cloud learning. In *AAAI*, pp. 3410–3418, 2025. doi:10.1609/aaai.v39i3.32353.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pp. 6840–6851, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Binbin Huang, Haobin Duan, Yiqun Zhao, Zibo Zhao, Yi Ma, and Shenghua Gao. CUPID: Generative 3d reconstruction via joint object and pose modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12741–12752, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Huang_CUPID_Generative_3D_Reconstruction_via_Joint_Object_and_Pose_Modeling_CVPR_2026_paper.pdf.

Zixuan Huang, Mark Boss, Aaryaman Vasishta, James M. Rehg, and Varun Jampani. SPAR3D: Stable point-aware reconstruction of 3d objects from single images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16860–16870, 2025. doi:10.1109/CVPR52734.2025.01571.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves, 2025. URL https://arxiv.org/abs/2505.02831. Accepted to ICLR 2026.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and Francois Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning*, pp. 5156–5165, 2020. URL https://proceedings.mlr.press/v119/katharopoulos20a.html.

Jae Joong Lee and Bedrich Benes. RGB2Point: 3d point cloud generation from single RGB images. In *WACV*, pp. 2952–2962, 2025. doi:10.1109/WACV61041.2025.00292.

Fei Li, Tiansong Li, Hongkui Wang, Kejun Wu, and Yu Li. MVP-Diff: Multi-view priors learning for diffusion-based single-view 3d point cloud reconstruction. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 12512–12516, 2026. doi:10.1109/ICASSP55912.2026.11461695.

Shaoming Li, Qing Cai, Songqi Kong, Runqing Tan, Heng Tong, Shiji Qiu, Yongguo Jiang, and Zhi Liu. MESC-3D: Mining effective semantic cues for 3d reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16912–16921, 2025a. doi:10.1109/CVPR52734.2025.01576.

Zhuoyuan Li, Yubo Ai, Jiahao Lu, ChuXin Wang, Jiacheng Deng, Hanzhi Chang, Yanzhe Liang, Wenfei Yang, Shifeng Zhang, and Tianzhu Zhang. Pamba: Enhancing global interaction in point clouds via state space model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5092–5100, 2025b. doi:10.1609/aaai.v39i5.32540.

Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. PointMamba: A simple state space model for point cloud analysis. In *Advances in Neural Information Processing Systems*, pp. 32653–32677, 2024. doi:10.52202/079017-1026.

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. doi:10.1109/ICCV51070.2023.00853.

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3D: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9970–9980, 2024. doi:10.1109/CVPR52733.2024.00951.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2837–2845, 2021. doi:10.1109/CVPR46437.2021.00286.

Luke Melas-Kyriazi, Christian Rupprecht, and Andrea Vedaldi. PC2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12923–12932, 2023. doi:10.1109/CVPR52729.2023.01242.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4460–4470, 2019. doi:10.1109/CVPR.2019.00459.

Shentong Mo. Scaling diffusion Mamba with bidirectional SSMs for efficient 3d shape generation. In *AAAI*, pp. 19475–19483, 2025. doi:10.1609/aaai.v39i18.34144.

Yuxuan Mu, Xinxin Zuo, Chuan Guo, Yilin Wang, Juwei Lu, Xiaofeng Wu, Songcen Xu, Peng Dai, Youliang Yan, and Li Cheng. GSD: View-guided gaussian splatting diffusion for 3d reconstruction. In *Proceedings of the European Conference on Computer Vision*, pp. 55–72, 2024. doi:10.1007/978-3-031-72986-7_4.

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts, 2022. URL https://arxiv.org/abs/2212.08751.

Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3DUnderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 55–64, 2020. doi:10.1109/CVPR42600.2020.00013.

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

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. RWKV: Reinventing RNNs for the transformer era. In *Findings of EMNLP 2023*, pp. 14048–14077, 2023. doi:10.18653/v1/2023.findings-emnlp.936. URL https://aclanthology.org/2023.findings-emnlp.936/.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017a. doi:10.1109/CVPR.2017.16.

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pp. 5099–5108, 2017b. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10881–10891, 2021. doi:10.1109/ICCV48922.2021.01072.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *Proceedings of the International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6550.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *Proceedings of the International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proceedings of the International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the International Conference on Machine Learning*, pp. 32211–32252, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B. Tenenbaum, and William T. Freeman. Pix3D: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2974–2983, 2018. doi:10.1109/CVPR.2018.00314.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models, 2023. URL https://arxiv.org/abs/2307.08621.

Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22819–22829, 2023. doi:10.1109/ICCV51070.2023.02086.

Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2Mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision*, pp. 52–67, 2018. doi:10.1007/978-3-030-01252-6_4.

Pengfei Wang, Liyi Chen, Zhiyuan Ma, Yanjun Guo, Guowen Zhang, and Lei Zhang. One2Scene: Geometric consistent explorable 3d scene generation from a single image. In *Proceedings of the International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.19766.

Tzu-Hsuan Weng, Robin Fischer, and Li-Chen Fu. 3d scene reconstruction from RGB images using dynamic graph convolution for augmented reality. In *IEEE VRW*, pp. 638–639, 2022. doi:10.1109/VRW55335.2022.00170.

Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2Vox: Context-aware 3d reconstruction from single and multi-view images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2690–2698, 2019. doi:10.1109/ICCV.2019.00278.

Haiyang Xu, Yu Lei, Zeyuan Chen, Xiang Zhang, Yue Zhao, Yilin Wang, and Zhuowen Tu. Bayesian diffusion models for 3d shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10628–10638, 2024. doi:10.1109/CVPR52733.2024.01011.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *Proceedings of the International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=DJSZGGZYVi.

Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. LION: Latent point diffusion models for 3d shape generation. In *Advances in Neural Information Processing Systems*, pp. 10021–10039, 2022. doi:10.52202/068431-0728.

Tao Zhang, Haobo Yuan, Lu Qi, Jiangning Zhang, Qianyu Zhou, Shunping Ji, Shuicheng Yan, and Xiangtai Li. Point cloud Mamba: Point cloud learning via state space model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10121–10130, 2025. doi:10.1609/aaai.v39i10.33098.

Chenliang Zhou, Fangcheng Zhong, Weihao Xia, Albert Miao, Canberk Baykal, and Cengiz Oztireli. Quartet of diffusions: Structure-aware point cloud generation through part and symmetry guidance. In *Proceedings of the International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=BT9rsod6Hc.

Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5826–5835, 2021. doi:10.1109/ICCV48922.2021.00577.

Wei Zhou, Xinzhe Shi, Xingxing Hao, Xing Hao, Kang Li, Jinye Peng, and Ying He. Point diffusion mamba: Unified diffusion-state-space modeling for single-view 3d reconstruction under data scarcity. In *Computer Vision – ECCV 2026*, Lecture Notes in Computer Science, pp. 131–150. Springer Nature Switzerland, 2026b. doi:10.1007/978-3-032-37624-4_8.

Yan Zhou, Dewang Ye, Huaidong Zhang, Xuemiao Xu, Huajie Sun, Yewen Xu, Xiangyu Liu, and Yuexia Zhou. Recurrent diffusion for 3d point cloud generation from a single image. *IEEE TIP*, 34:1753–1765, 2025. doi:10.1109/TIP.2025.3539935.

Dekai Zhu, Yan Di, Stefan Gavranovic, and Slobodan Ilic. SeaLion: Semantic part-aware latent point diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11789–11798, 2025. doi:10.1109/CVPR52734.2025.01101.

SUPPLEMENTARY CONTENTS

Appendix order.

- Section A: Highlights.
- Section B: supplementary quantitative results are ordered for review as ShapeNet-R2N2 (Secs. B.2–B.3), Pix3D (Sec. B.4), CO3Dv2 (Secs. B.5–B.6), and Objaverse (Sec. B.7); inference efficiency and recent-method comparisons follow, then feature-depth, representation-guidance, serialization, StateEvo, retention, scan-direction, and backbone ablations (Secs. B.10–B.12), with exact EMD@4096 comparisons placed last (Sec. B.13). The evaluation protocol and baseline scope are introduced at the start of the section (Sec. B.1).
- Section C: theoretical analysis of the StateEvo recurrence, including its local objective, retention bound, exact chunkwise form, linear sequence complexity, and tied-gate reduction.
- Sections D–F: comparisons with related sequence mixers, diffusion preliminaries, quantitative metrics, computing infrastructure, training hyperparameters, and RasterProj implementation details.
- Section G: Limitations and Broader Impact.
- Section H: qualitative ShapeNet-R2N2, Pix3D, and CO3Dv2 visualizations.

The quantitative sections use a common evaluation convention: predictions and references contain 4,096 points, are independently centroid aligned, and are scored with the same CD, F1, and exact EMD@4096 implementations. Unless stated otherwise, quality results use 1,000-step DDPM sampling with the linear beta schedule described in Section F. The efficiency section distinguishes the averaged controlled profile from the one-repeat NFE diagnostic so that runtime and memory quantities retain their measurement definitions.

A HIGHLIGHTS

The main highlights of RepStateDiff are summarized as follows:

- **State-evolution denoising.** RepStateDiff separates persistent shape retention from image-conditioned evidence updates through StateEvo, whose erase, write, and decay gates maintain a coherent evolving point-cloud state.
- **Representation-guided conditioning.** Frozen visual features are aligned to noisy 3D points and further coupled to internal denoising through intermediate representation alignment and internal guidance.
- **Low-data reconstruction.** On the primary ShapeNet-R2N2 categories, the method improves average CD and F1-score over the recent PDM baseline at 10% data; the remaining categories are compared with the available earlier baselines.
- **Quality-efficiency balance.** RepStateDiff combines a compact 38.11M-parameter model with a favorable controlled runtime and peak-memory profile at the 4,096-point evaluation resolution.

The architectural roles are hierarchical rather than interchangeable: StateEvo provides the central denoising state transition, surface projection and multi-layer visual features supply image-aligned conditions, and intermediate alignment plus internal guidance couple those conditions to the denoising representation. The local detail refiner is a supporting residual module for visible-surface correction; the reported ablations do not assign it a separate stand-alone contribution estimate.

B SUPPLEMENTARY EXPERIMENTAL ANALYSIS

This section extends the quantitative and qualitative analysis in the main paper. The appendix follows the ICLR two-column format and uses single-column tables whenever the comparison can be read locally.

Dataset split statistics. Tables 6 and 7 summarize the train/test split statistics used for the supplementary category-level evaluation. These statistics follow the RGF appendix split table and are included here to make the low-data and category-level comparisons easier to audit.

Table 6: ShapeNet-R2N2 dataset split statistics.

Category	Train	Test	Total	Category	Train	Test	Total
Airplane	3236	809	4045	Lamp	1854	464	2318
Bench	1452	363	1815	Loudspeaker	1292	323	1615
Cabinet	1257	315	1572	Rifle	1898	475	2373
Car	5997	1500	7497	Sofa	2538	635	3173
Chair	5420	1355	6775	Table	6800	1700	8500
Display	876	219	1095	Telephone	841	211	1052
				Watercraft	1550	388	1938

Table 7: Pix3D dataset split statistics for the categories used in the full-data evaluation.

Category	Train	Test	Total
Chair	3071	768	3839
Sofa	1557	390	1947
Table	1496	374	1870

B.1 SCOPE OF QUANTITATIVE BASELINES

The numerical comparison is restricted to direct point-set systems evaluated on compatible ShapeNet-R2N2 or Pix3D splits, coordinate normalization, and 4,096-point CD/F1 protocols. PC2, CCD-3DR, BDM, MESC-3D, and PDM provide published results under the compatible protocols used in the main tables Melas-Kyriazi et al. (2023); Di et al. (2023); Xu et al. (2024); Li et al. (2025a); Zhou et al. (2026b); PDM is the recent diffusion–state-space baseline. Its category-level results cover the three primary ShapeNet-R2N2 and three main Pix3D categories, but not the remaining ShapeNet-R2N2 and Pix3D categories analyzed below. Mesh, implicit, radiance-field, and Gaussian systems are not numerically mixed because their output conversion, camera assumptions, and test-time procedures are different.

Cross-representation scope. A common point-set metric also requires a common output construction. Sampling 4,096 points from a mesh or implicit surface, extracting an iso-surface, or aggregating Gaussian primitives introduces representation-specific choices that can change CD, F1, and EMD independently of the original method. Some structured or latent systems additionally use multi-view priors, extra pretraining data, or test-time optimization. The quantitative tables are therefore intended to establish protocol-matched behavior among direct point-set systems, while broader single-image 3D methods remain part of the related-work context rather than being assigned a numerical rank in the point-cloud comparison.

Protocol compatibility, rather than output novelty, determines inclusion. Other direct point-output methods such as RGB2Point Lee & Benes (2025) and one-step mean-flow Point-MF Baba & Yanai (2026) are useful efficiency references, but their objectives, point counts, samplers, or evaluation settings are not interchangeable with the 1,000-step DDPM protocol. They are therefore discussed as complementary methods rather than assigned a numerical rank.

The controlled Mamba/RWKV comparison replaces only the sequence-mixing block in the same hierarchical denoiser; it is not a reimplementation of PointMamba, PointRWKV, or TFDM Liang et al. (2024); He et al. (2025); Mo (2025). The variants are not strictly matched in parameter count or FLOPs, so this is a block-level control under a shared reconstruction protocol rather than a compute-normalized architecture comparison. Within the shared point-set protocol, PC2 adds projection conditioning, CCD-3DR enforces conditioning consistency, BDM models uncertainty, and RepStateDiff changes the denoising state dynamics. The external tables therefore compare complete systems, while the mixer control isolates the block-level contribution.

PC2 provides a representative projection-conditioned formulation. In the reported RepStateDiff configuration, surface projection supplies RGB and frozen visual features indexed by the current noisy 3D points; these features then enter hierarchical token construction, representation alignment, and the StateEvo update. The distinction is therefore the coupling of point-indexed visual conditioning to state-evolution denoising, rather than surface rasterization itself.

Low-data interpretation. The 1% setting is evaluated quantitatively on all three primary ShapeNet-R2N2 categories. Under this budget, RepStateDiff improves F1-score in all three categories and obtains the lowest CD on chair, whereas PDM remains better on CD for airplane and car Zhou et al. (2026b). These results support improved low-data behavior under the reported category-specific protocol without implying uniform dominance across all metrics. The tabulated 1% values are point estimates from the fixed subset and seed-controlled training configuration; variation across alternative subsets or training seeds is not measured here.

B.2 SHAPENET-R2N2 CATEGORY-LEVEL ANALYSIS

Relative improvement on the main categories. Tables 8 convert the main ShapeNet-R2N2 results into relative CD reduction and F1 increase for the three primary categories and all three training-data budgets.

Table 8: Relative improvement on the three primary ShapeNet-R2N2 categories.

Category	Comparator	10%		50%		100%	
		CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
Chair	vs. PC2	+33.20%	+22.39%	**+31.37%**	+20.14%	+29.27%	+16.59%
	vs. CCD-3DR	+27.65%	+15.07%	+20.01%	+10.76%	+20.68%	+8.63%
	vs. BDM-M	+31.58%	+21.77%	+29.43%	+17.71%	+28.07%	+15.60%
	vs. BDM-B	+31.38%	+17.32%	+27.85%	+13.39%	+27.77%	+11.55%
	vs. PDM	+21.17%	+14.80%	+26.65%	+12.90%	+25.36%	+10.86%
	vs. MESC-3D	**+36.01%**	**+26.25%**	+31.26%	**+22.95%**	**+29.40%**	**+18.12%**
Airplane	vs. PC2	**+42.43%**	+14.21%	**+46.93%**	+16.56%	**+43.08%**	+13.44%
	vs. CCD-3DR	+37.68%	+12.91%	+44.05%	+15.28%	+40.18%	**+14.13%**
	vs. BDM-M	+42.27%	**+14.40%**	+44.61%	+16.38%	+42.37%	+13.78%
	vs. BDM-B	+39.42%	+12.91%	+40.96%	+14.20%	+36.40%	+12.58%
	vs. PDM	+16.46%	+12.54%	+19.15%	+10.41%	+22.83%	+3.34%
	vs. MESC-3D	+31.83%	+13.09%	+20.94%	**+18.26%**	+25.70%	+4.06%
Car	vs. PC2	**+17.80%**	+9.73%	**+18.77%**	+12.55%	+22.14%	**+12.98%**
	vs. CCD-3DR	+15.38%	+8.29%	+18.33%	+10.91%	+19.02%	+9.96%
	vs. BDM-M	+15.91%	+9.73%	+16.26%	+11.11%	+21.90%	+11.55%
	vs. BDM-B	+11.67%	+6.68%	+18.76%	+10.11%	**+23.90%**	+10.55%
	vs. PDM	+3.28%	+6.09%	+8.45%	+9.32%	+2.77%	+1.81%
	vs. MESC-3D	+5.08%	**+10.15%**	–	**+14.66%**	+3.62%	+2.66%

The merged view shows the strongest and most consistent gains on chair and airplane, while car has smaller CD margins but still consistent F1 improvements. The PDM rows provide the current direct diffusion comparison for these primary categories; the remaining rows preserve earlier published baselines.

Remaining ten ShapeNet-R2N2 categories at 50% and 100% data. The following tables extend the remaining-category analysis to the 50% and 100% training-data regimes. They use the same 4,096-point output and metric scaling as the main ShapeNet-R2N2 results: CD is multiplied by 10^3, EMD by 10^2, and higher F1 is better.

Table 9: RepStateDiff on the remaining ten ShapeNet-R2N2 categories at 50% and 100% training data. CD is multiplied by 10^3 and EMD by 10^2.

	50% training data			100% training data		
Category	CD↓	F1↑	EMD↓	CD↓	F1↑	EMD↓
Bench	39.67	0.7014	20.3366	35.48	0.7372	18.7068
Cabinet	44.49	0.5035	22.8086	39.89	0.5464	21.4948
Display	71.24	0.5828	25.7552	60.45	0.6053	24.1575
Lamp	116.49	0.4983	36.3387	98.81	0.5206	34.0785
Loudspeaker	71.57	0.4305	27.7488	68.38	0.4504	26.6094
Rifle	24.78	0.8098	22.7089	22.95	0.8231	21.4810
Sofa	41.11	0.5300	22.3945	38.58	0.5417	21.5162
Table	55.67	0.6109	24.5187	49.28	0.6339	23.2013
Telephone	42.22	0.6803	19.9318	36.36	0.7031	18.5960
Watercraft	38.48	0.6307	23.1315	36.43	0.6381	22.1782
10-category mean	**54.57**	**0.5978**	**24.5673**	**48.66**	**0.6200**	**23.2020**

The additional regimes show the expected data-scaling trend: the mean CD decreases from 54.57 at 50% data to 48.66 at 100%, while mean F1 rises from 0.5978 to 0.6200 and mean EMD decreases from 24.5673 to 23.2020.

B.3 Joint 13-Category Training

Category-specific models isolate within-category reconstruction, but they do not test whether one model can share statistical strength across object types. We therefore define a category-shared experiment in which a single RepStateDiff model is trained on the union of deterministic 10% subsets from all 13 ShapeNet-R2N2 categories and evaluated on the same full per-category evaluable sets as the category-specific models. The fixed loader intersects the nominal test split with objects having paired point clouds, yielding 7,806 evaluable instances in total; both settings use identical per-category files. All categories are observed during training, so this experiment measures joint seen-category training rather than unseen-category generalization. It uses the same 4,096-point output and CD, F1, and exact EMD@4096 protocols as the category-specific runs.

Table 10: Category-specific and joint 13-category RepStateDiff models on ShapeNet-R2N2. Both settings use 10% of each category's training split and identical evaluable test instances; the joint setting trains one shared model. CD and exact EMD@4096 are multiplied by 10^3 and 10^2, respectively. Macro averages are computed from the displayed category-level values.

	Category-specific			Joint 13-category		
Category	CD↓	F1↑	EMD↓	CD↓	F1↑	EMD↓
Airplane	50.66	0.6910	29.14	**32.00**	**0.7720**	**23.57**
Bench	**60.23**	**0.5990**	**25.85**	64.85	0.5480	28.78
Cabinet	**59.55**	**0.4320**	**26.52**	83.66	0.3380	31.50
Car	**53.42**	**0.5750**	**24.80**	53.43	0.5600	26.84
Chair	**64.96**	**0.4810**	**28.97**	75.05	0.4470	30.97
Display	**77.71**	**0.5380**	**29.17**	102.01	0.4490	33.88
Lamp	**137.40**	**0.4350**	**41.11**	162.38	0.3920	44.49
Loudspeaker	**92.11**	**0.3730**	**32.11**	125.05	0.2960	37.57
Rifle	37.83	0.7730	26.78	**28.19**	**0.7980**	**24.93**
Sofa	**52.16**	**0.4770**	**25.99**	65.42	0.4200	30.03
Table	**67.50**	**0.5580**	**27.34**	81.19	0.5170	30.50
Telephone	**50.62**	**0.6270**	**22.09**	89.40	0.4400	30.14
Watercraft	**48.23**	0.5810	**26.65**	49.82	**0.5950**	28.14
Macro average	**65.57**	**0.5490**	**28.19**	77.88	0.5060	30.87

The joint run uses the pooled 10%-per-category subset rather than 10% of a category-agnostic shuffled list, preserving representation from every class. It is trained for 50,000 optimization steps with a learning rate of 1.0×10^{-3} and batch size 16, and sampled with 1,000 DDPM steps. The shared model improves both CD and F1 on airplane and rifle, and also improves watercraft F1. Because 50,000 pooled updates are not exposure matched to training 13 separate category-specific models, this table is a practical shared-model comparison rather than a controlled claim about unseen-category generalization.

Relative improvements on the remaining ShapeNet-R2N2 categories. Table 11 uses the same split as the absolute-result table and reports relative CD reduction and F1 increase over BDM-B and MESC-3D for all ten remaining categories.

Table 11: Relative improvement on the remaining ten ShapeNet-R2N2 categories under 10% data. CD entries are relative reductions and F1 entries are relative increases.

Category	Comparator	CD reduction↑	F1 increase↑
Bench	vs. BDM-B	+25.58%	+10.47%
	vs. MESC-3D	**+66.61%**	**+77.67%**
Cabinet	vs. BDM-B	+39.22%	+26.57%
	vs. MESC-3D	**+70.68%**	**+72.64%**
Display	vs. BDM-B	+34.87%	+3.71%
	vs. MESC-3D	**+55.74%**	**+41.27%**
Lamp	vs. BDM-B	+35.05%	+27.44%
	vs. MESC-3D	**+35.45%**	**+87.32%**
Loudspeaker	vs. BDM-B	+22.51%	+11.87%
	vs. MESC-3D	**+61.64%**	**+50.82%**
Rifle	vs. BDM-B	+4.85%	+2.20%
	vs. MESC-3D	**+59.23%**	**+6.28%**
Sofa	vs. BDM-B	+20.23%	+8.82%
	vs. MESC-3D	**+68.70%**	**+44.88%**
Table	vs. BDM-B	+46.60%	+28.83%
	vs. MESC-3D	**+71.85%**	**+157.07%**
Telephone	vs. BDM-B	+19.62%	–
	vs. MESC-3D	**+64.87%**	**+27.95%**
Watercraft	vs. BDM-B	+17.34%	+4.92%
	vs. MESC-3D	**+67.63%**	**+37.41%**

The relative-improvement view compares the ten categories with the available BDM-B and MESC-3D reports; PDM does not report these category-level results. RepStateDiff improves CD over BDM-B on all ten categories and shows substantial F1 gains across the group.

B.4 Pix3D Category-Level Analysis

Full-data relative improvement. Pix3D evaluates reconstruction from real images, where backgrounds and object appearance are less controlled than rendered ShapeNet-R2N2 views. Table 12 summarizes the full-data relative improvements for the three Pix3D categories reported in the main paper. Figure 6 complements the category-level tables with matched qualitative comparisons under the 1%, 10%, and 100% training-data settings.

Table 12: Relative improvement over the main Pix3D comparators under 100% training data.

Comparator	Chair		Sofa		Table	
	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
vs. PC2	**+50.12%**	+34.54%	**+11.55%**	+5.39%	+24.17%	+9.82%
vs. CCD-3DR	+48.10%	+30.70%	+7.10%	+4.22%	+21.66%	+4.31%
vs. BDM-M	+49.00%	+32.74%	+6.25%	+3.99%	+23.91%	+5.57%
vs. BDM-B	+47.71%	+30.99%	+7.39%	+3.08%	+17.53%	+1.63%
vs. PDM	+27.06%	+19.44%	-0.70%	+1.30%	+16.59%	+3.32%
vs. MESC-3D	+36.70%	**+61.08%**	+0.62%	**+65.14%**	**+25.41%**	**+42.48%**

Relative to PDM, the gains are largest on chair and table; sofa is a tighter category in CD, where PDM remains slightly better. The older BDM-B and MESC-3D rows provide additional historical context for the full-data comparison.

Extended Pix3D results under 10% training data. Table 13 combines the category-level absolute and relative views beyond the three Pix3D categories used in the main paper. The MESC-3D, BDM-B, and RepStateDiff entries are extracted from the category-wise evaluation logs.

Relative improvements under 10% training data. The lower block of Table 13 reports relative CD reduction and F1 increase for the same categories.

Table 13: Pix3D category-level results under 10% training data. The upper block reports absolute CD and F1; the lower block reports relative CD reduction and F1 increase.

Absolute metrics (CD↓, F1↑)

Method	Bed		Bookcase		Chair	
	CD	F1	CD	F1	CD	F1
BDM-B	151.69	0.5640	**61.91**	**0.5570**	73.35	0.5590
MESC-3D	131.80	0.2980	135.06	0.3190	165.18	0.2760
RepStateDiff	**119.81**	**0.5900**	67.16	0.5440	**53.22**	**0.6110**

Method	Sofa		Table		Wardrobe	
	CD	F1	CD	F1	CD	F1
BDM-B	41.72	0.4880	236.27	**0.3710**	133.99	**0.2830**
MESC-3D	63.56	0.4060	361.43	0.2360	198.83	0.2470
RepStateDiff	**37.92**	**0.5060**	**212.04**	0.3460	**125.32**	0.2800

Relative improvements (CD reduction↑, F1 increase↑)

Comparator	Bed		Bookcase		Chair		Desk	
	CD	F1	CD	F1	CD	F1	CD	F1
vs. BDM-B	**+21.02%**	+4.64%	–	–	+27.44%	+9.26%	–	–
vs. MESC-3D	+9.10%	**+97.99%**	**+50.27%**	**+70.53%**	**+67.78%**	**+121.38%**	–	**+27.97%**

Comparator	Sofa		Table		Wardrobe	
	CD	F1	CD	F1	CD	F1
vs. BDM-B	+9.11%	+3.59%	+10.26%	–	+6.47%	–
vs. MESC-3D	**+40.34%**	**+24.63%**	**+41.33%**	**+46.61%**	**+36.97%**	**+13.36%**

Together, the absolute and relative Pix3D tables show that RepStateDiff improves most CD metrics and several F1 metrics under 10% Pix3D training data, providing evidence for stronger real-image geometric proximity.

B.5 CO3Dv2 Evaluation Across Training-Data Regimes

We further evaluate RepStateDiff on the CO3Dv2 real-image dataset Reizenstein et al. (2021) using the *hydrant* and *teddybear* categories. For each split in `fewview_dev`, we first select one deterministic frame per sequence and then construct seeded 10%, 50%, and 100% subsets from the resulting training set; the validation set remains fixed across all three regimes. The 10%, 50%, and 100% models are trained for 10,000, 50,000, and 100,000 steps with learning rates of 1.0×10^{-3}, 5.0×10^{-4}, and 3.0×10^{-4}, respectively. Evaluation covers all 50 *hydrant* and 93 *teddybear* validation instances, with every prediction–reference pair evaluated successfully. Each reconstruction contains 4,096 points. CD and F1-score independently align the prediction and reference centroids, F1-score uses a squared-distance threshold of 0.01, and EMD uses exact linear-sum assignment over all 4,096 points.

Table 14: CO3Dv2 reconstruction across category-specific training-data regimes. CD is multiplied by 10^3 and EMD by 10^2; lower CD/EMD and higher F1 are better. Light-red cells mark the best regime within each category.

Category	Training data	CD↓	F1↑	EMD↓
Hydrant	10%	141.92	0.3960	39.91
	50%	111.69	0.5005	31.92
	100%	**92.53**	**0.5395**	**28.38**
Teddybear	10%	177.87	0.3000	49.62
	50%	103.13	0.3744	37.04
	100%	**95.28**	**0.3946**	**33.89**

Table 14 shows monotonic improvement in all three metrics as category-specific supervision increases. From 10% to 100% training data, *hydrant* CD decreases from 141.92 to **92.53**, F1-score rises from 0.3960 to **0.5395**, and EMD decreases from 39.91 to **28.38**. For *teddybear*, CD decreases from 177.87 to **95.28**, F1-score rises from 0.3000 to **0.3946**, and EMD decreases from 49.62 to **33.89**. The largest change occurs between 10% and 50% data for *teddybear*, while the 100% models

still improve every metric for both categories. Because this experiment does not include a protocol-matched external baseline, it should be interpreted as a within-model data-scaling diagnostic on CO3Dv2 rather than a cross-method ranking.

B.6 CO3Dv2 13-Category Full-Data Evaluation

To broaden the real-image evaluation, we report the 100% training-data results for all 13 CO3Dv2 categories available in the evaluation summary. The effective-sample count is the number of validation instances with a successful prediction–reference pair. All reconstructions contain 4,096 points, and CD and EMD use the scaling defined in Section E.2.

Table 15: CO3Dv2 reconstruction on 13 categories with 100% training data. CD is multiplied by 10^3 and EMD by 10^2.

Category	Samples	CD↓	F1↑	EMD↓
Backpack	106	129.06	0.3658	36.3788
Book	83	105.30	0.4503	29.7932
Cellphone	50	218.17	0.4521	33.7006
Chair	88	140.83	0.4826	34.8203
Hydrant	50	92.53	0.5395	28.3803
Keyboard	81	138.13	0.5389	26.6872
Mouse	50	73.95	0.4735	31.3282
Plant	78	193.03	0.3113	45.5627
Remote	50	70.22	0.5647	24.6022
Teddybear	93	95.28	0.3946	33.8904
Toilet	33	206.53	0.4278	44.7322
Umbrella	50	109.65	0.5266	28.7998
Vase	86	71.14	0.4738	31.2482
13-category mean / total	898	**126.45**	**0.4617**	**33.0711**

The 13-category mean provides a broader real-image reference point than the two-category data-scaling diagnostic above. The category spread is substantial, with the lowest CD on vase and the highest F1 on remote, which underscores the diversity of object geometry and appearance in CO3Dv2.

B.7 Objaverse 12-Category Full-Data Evaluation

We additionally evaluate the 100% training-data models on a 12-category Objaverse subset. The table reports the same 4,096-point reconstruction metrics and scaling used throughout the appendix; the final row is the unweighted macro average across categories.

Table 16: Objaverse reconstruction on 12 categories with 100% training data. CD is multiplied by 10^3 and EMD by 10^2.

Category	CD↓	F1↑	EMD↓
Airplane	133.38	0.5432	41.272387
Antenna	285.12	0.4274	52.399351
Car	100.05	0.3766	40.569690
Chair	57.63	0.7287	19.891548
Chandelier	82.91	0.4350	32.500366
Cone	146.12	0.5567	31.370365
Control	398.42	0.4477	41.892427
Gravestone	130.68	0.3462	38.139215
Seashell	173.00	0.2984	40.140585
Shield	238.83	0.4959	40.443677
Snowman	49.90	0.6131	21.082747
Sword	41.03	0.8087	32.201281
Macro average	**153.09**	**0.5065**	**35.991970**

The Objaverse results show broad category variation: the sword and chair categories obtain the highest F1 scores, whereas control and antenna are more challenging in CD. These values are reported as a full-data diagnostic and are not merged into the ShapeNet-R2N2 or CO3Dv2 rankings because the datasets use different image and geometry distributions.

Table 17: Runtime–NFE and traceable-operator FLOPs profile for 4,096-point sampling with batch size 1. Repeated configurations shared among ablation tables are listed once.

		Traceable	Runtime (s / sample)				
Family	Configuration	GFLOPs@1,000	NFE 50	NFE 100	NFE 250	NFE 500	NFE 1,000
Reference	Full model	74,678.8	2.30	5.17	11.62	24.82	48.02
Scale	Single-scale $[256]$	40,765.5	1.04	2.05	5.13	10.24	20.48
Scale	Three-scale $[512, 256, 128]$	52,592.3	1.44	2.87	7.18	14.33	28.59
Scale	Single-scale $[512]$	51,025.1	1.15	2.29	5.72	11.33	22.77
Scale	Single-scale $[1024]$	71,544.5	1.36	2.70	6.75	13.40	26.92
Scale	Three-scale $[2048, 1024, 512]$	118,851.8	2.78	5.54	13.80	27.71	55.25
Serialization	Coordinate-sum sorting, distance on	74,678.8	2.01	5.25	10.50	20.78	41.09
Serialization	Morton (Z-order), distance on	74,678.8	2.17	4.29	10.91	21.83	43.39
Serialization	Native FPS traversal, distance off	74,678.8	1.94	3.90	9.42	18.43	37.42
Serialization	Coordinate-sum sorting, distance off	74,678.8	1.88	3.75	9.37	18.65	37.26
Serialization	Morton (Z-order), distance off	74,678.8	2.03	4.06	10.07	20.16	40.39
Order sensitivity	FPS-trained, coordinate-sum sampling	74,678.8	1.88	3.75	9.31	18.55	37.19
Order sensitivity	FPS-trained, Morton sampling	74,678.8	2.03	4.05	10.09	20.29	40.58
Decay	Free initialization, all modulation on	74,678.8	1.83	3.66	9.12	18.30	36.59
Decay	Without time modulation	74,678.8	1.83	3.64	9.12	18.20	36.31
Decay	Without distance/time, log-spaced initialization	74,678.8	1.83	3.61	8.99	18.07	36.19
Decay	Without distance/time, free initialization	74,678.8	2.80	4.78	9.89	19.92	40.22
Decay	Token-invariant decay	74,141.9	1.80	3.58	8.95	17.89	35.69
Scan	Forward only	74,678.8	1.67	3.44	8.51	16.95	33.96
Scan	Independent bidirectional average	74,678.8	1.85	3.70	9.24	18.49	36.90
Visual teacher	MSN ViT-S/16	74,676.8	1.85	3.70	9.23	18.49	37.12
Visual teacher	MAE ViT-B/16	100,566.9	1.95	3.88	9.72	19.33	38.83
Category sharing	Joint 13-category model	74,678.8	1.84	3.68	9.15	18.40	36.75

B.8 Inference Efficiency Across Point-Cloud Resolutions

We further evaluate inference efficiency while varying the point-cloud resolution from 1,024 to 4,096 points. This controlled benchmark complements the aggregate efficiency comparison in the main paper by reporting runtime variation and separating absolute peak memory from the incremental memory allocated during sampling. Tables 18 and 19 report the absolute measurements and the corresponding relative comparison, respectively.

Table 18: Absolute inference efficiency under different point-cloud resolutions. Runtime is reported as mean $\pm$ standard deviation over 50 measured samples after five warm-up samples. Peak and incremental memory report PyTorch-allocated GPU memory. All methods use batch size 1, FP32 precision, and 1,000 DDPM sampling steps on an NVIDIA RTX 3090.

Points	Method (M)	Time (s)	Peak (GiB)	Increment (GiB)
1,024	RepStateDiff (38.11)	**19.789 $\pm$ 0.043**	0.338	0.188
1,024	PC2 (49.71)	44.578 $\pm$ 0.033	**0.334**	**0.147**
1,024	BDM-B (77.36)	47.480 $\pm$ 0.119	0.437	**0.147**
2,048	RepStateDiff (38.11)	**23.932 $\pm$ 0.045**	0.356	0.206
2,048	PC2 (49.71)	48.187 $\pm$ 0.039	**0.336**	**0.149**
2,048	BDM-B (77.36)	51.247 $\pm$ 0.035	0.439	**0.149**
4,096	RepStateDiff (38.11)	**36.347 $\pm$ 0.099**	0.394	0.244
4,096	PC2 (49.71)	55.596 $\pm$ 0.050	**0.339**	**0.152**
4,096	BDM-B (77.36)	59.280 $\pm$ 0.051	0.442	**0.152**

In this table, "Peak" denotes the maximum PyTorch-allocated GPU memory observed during a measured sample, whereas "Increment" denotes the peak allocation minus the allocation recorded immediately before sampling. The 4,096-point resolution-specific RepStateDiff diagnostic is therefore 36.347$\pm$0.099 s with 0.394 GiB absolute peak allocation and 0.244 GiB incremental allocation. The headline efficiency table in the main paper uses a separate aggregate profile: 38.86 s per object and 0.41 GB additional peak sampling memory for RepStateDiff. The latter is the quantity used for the headline comparison and should not be conflated with the absolute-peak and incremental values in this resolution diagnostic.

Table 19: Relative efficiency of RepStateDiff against PC2 and BDM-B. RepStateDiff contains 23.34% fewer parameters than PC2 and 50.74% fewer parameters than BDM-B. Relative values are computed from the unrounded measurements.

Points	Comparator	Time reduction	Speedup	Peak-memory reduction
1,024	PC2	55.61%	2.253×	–
1,024	BDM-B	**58.32%**	2.399×	**22.67%**
2,048	PC2	50.33%	2.013×	–
2,048	BDM-B	**53.30%**	2.141×	**18.92%**
4,096	PC2	34.62%	1.530×	–
4,096	BDM-B	**38.69%**	1.631×	**10.89%**

At 1,024–4,096 points, RepStateDiff is consistently faster than PC2 and BDM-B, with the largest speedups at the lowest resolution. Its peak memory also remains below BDM-B through 4,096 points. These results show a favorable parameter and runtime trade-off up to 4,096 points, the resolution used in the main experiments.

Runtime–NFE and FLOPs profile. The NFE sweep in Table 17 is a one-repeat diagnostic profile recorded at batch size one for NFE $\{50, 100, 250, 500, 1000\}$. It uses a 10-step warm-up for each NFE and is reported to expose the scaling trend; the averaged five-warm-up/50-sample profile above is the reference for the main efficiency table. FLOPs count PyTorch-registered, traceable operators; unsupported custom kernels and PyTorch3D operators are excluded and are not presented as total end-to-end FLOPs.

The 4,096-point value in the resolution-specific table and the 4,096-point value in the headline efficiency table are profile-specific quantities: the former reports the averaged controlled measurement of 36.347±0.099 s, whereas the latter reports the separate aggregate profile of 38.86 s. They are therefore not interchangeable measurements. Likewise, the NFE$=1{,}000$ entry in the one-repeat table is a scaling diagnostic rather than a replacement for either headline profile. Runtime claims are interpreted only with the measurement protocol named in the corresponding table.

The headline efficiency table is a fixed-profile comparison using the reported baseline measurements and their documented inference settings; it is not a quality–latency Pareto frontier obtained by retuning every external method. RepStateDiff uses the linear-beta DDPM scheduler described in Section F, while the available baseline reports do not establish that every method uses the same scheduler, conditioning backbone, or internal evaluation pipeline. The NFE and runtime diagnostics therefore make the measurement definitions explicit without implying a universal NFE-normalized ranking across all generative models.

For RepStateDiff, the timed sampling window includes the frozen DINOv2 forward pass, conditioning-map construction, all RasterProj calls, the complete DDPM trajectory, local detail refinement, and output conversion. The frozen encoder is evaluated once per input and its feature map is reused across reverse steps; the reported runtime therefore includes feature extraction even though its weights are not updated.

B.9 Recent Diffusion-Based Comparisons

MVP-Diff on ShapeNet-R2N2. MVP-Diff Li et al. (2026) is a multi-view-prior-assisted approach to single-view reconstruction. Although reconstruction is conditioned on one image, its prior-learning stage processes six views of each object, uses their camera extrinsics, and aligns the inferred tokens with features from the remaining views. RepStateDiff instead trains directly from single-view image–shape pairs and requires neither cross-view feature targets nor multi-view extrinsic supervision. We therefore treat MVP-Diff as a complementary multi-view-prior baseline rather than a strictly protocol-matched single-view baseline. Table 20 compares the methods on the three ShapeNet-R2N2 categories shared by the reported results.

Table 20: ShapeNet-R2N2 comparison with MVP-Diff. CD is multiplied by 10^3.

Method	Chair		Airplane		Car	
	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
PC2	65.57	0.4640	65.97	0.6550	64.36	0.5470
BDM-B	64.21	0.4850	59.04	0.6600	65.85	0.5590
PDM	62.14	0.4880	48.66	0.7190	51.54	0.6070
MESC-3D	65.69	0.4850	50.54	0.7140	51.99	0.6020
MVP-Diff	48.04	**0.5910**	**30.07**	**0.7870**	**25.15**	**0.7080**
RepStateDiff	**46.38**	0.5410	37.55	0.7430	50.11	0.6180

Table 21: Pix3D comparison with MVP-Diff. CD is multiplied by 10^3.

Method	Chair		Sofa		Table	
	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
PC2	115.94	0.4430	47.17	0.4450	202.77	0.3970
BDM-B	110.60	0.4550	45.05	0.4550	186.46	0.4290
PDM	79.28	0.4990	41.43	0.4630	184.35	0.4220
MESC-3D	91.36	0.3700	41.98	0.2840	206.16	0.3060
MVP-Diff	104.38	0.4940	**35.54**	**0.5080**	**115.10**	**0.4780**
RepStateDiff	**57.83**	**0.5960**	41.72	0.4690	153.77	0.4360

The reported MVP-Diff setup uses four RTX A6000 GPUs. The tables include PDM as the current direct baseline for the shared primary categories, while MVP-Diff remains a contextual multi-view-prior comparison. MVP-L is pretrained for 3,000 epochs with batch size 512, while MVP-Diff is trained for 500,000 steps with batch size 96. Its MVP-E module uses a 12-layer ViT-Base over six views, and its diffusion backbone contains four 768-dimensional, eight-head WAFBs. RepStateDiff is trained on one RTX 3090. These differences in hardware, training schedule, and multi-view prior construction preclude a compute-matched rerun under the present single-GPU protocol. The two tables therefore report the published MVP-Diff values as contextual comparisons. Under these different protocols, RepStateDiff achieves a lower ShapeNet chair CD (46.38 versus 48.04) without multi-view prior learning, while MVP-Diff obtains the higher chair F1-score (0.5910 versus 0.5410). On Pix3D, RepStateDiff gives the best chair CD and F1, demonstrating effective single-image point-state refinement on real images.

B.10 Image-Feature Layer-Depth Diagnostic

The integration of multi-layer visual features yields consistent performance gains. When combined with StateEvo, they reduce the mean CD from 61.94 to **59.06** and increase the mean F1-score from 0.5611 to **0.5684**. Table 22 further isolates this design choice on the *chair* category: increasing the number of aggregated image-feature layers from one to four steadily reduces the CD from 75.61 to **64.96** and increases the F1-score from 0.4581 to **0.4814**. These results suggest that lower-level features provide essential local surface cues, while higher-level features contribute broader semantic context.

Table 22: Image-feature layer-depth diagnostic on 10% ShapeNet-R2N2 chair. Aggregating four image-feature layers gives the lowest CD (64.96) and highest F1-score (0.4814). The monotonic improvement from one to four layers supports multi-layer visual aggregation.

layers	CD↓	F1↑
1 layer	75.61	0.4581
2 layers	70.23	0.4681
3 layers	67.86	0.4725
4 layers	**64.96**	**0.4814**

B.11 Representation-Guidance Interaction

The two representation-guided components operate at different points in the denoising computation. Intermediate alignment constrains decoded point features to remain close to the projected visual representation, whereas internal guidance uses a weak noise head attached to the same decoded features. The weak head is trained to predict the injected noise from the intermediate state, and the sampling correction is formed from the residual between the full prediction and this auxiliary prediction. Consequently, the correction is zero when the two predictions agree and increases only along a direction supplied by the conditioned denoiser itself.

The ablation in Table 4 shows that this interaction is conditional on the richness of the visual features. StateEvo has mean CD/F1 of 61.94/0.5611, and adding multi-layer visual features changes these values to 59.06/0.5684. Internal guidance then gives 56.45/0.5818, while adding intermediate alignment as well gives the full model's 56.34/0.5824. In contrast, applying guidance without the multi-layer features gives a mean CD of 63.49, higher than the StateEvo value. The result is consistent with a representation correction that benefits from complementary local and semantic visual

cues rather than acting as an unconditional regularizer. The reported configuration uses $s_{\text{ig}} = 0.5$, with both auxiliary loss weights set to 0.1 (Section F).

For a compact reading of the full ablation, the Base configuration retains the shared denoiser, tokenization, projection, and diffusion objective while leaving S, M, A, and G disabled. Relative to Base, the successive S, S+M, S+M+G, and Full rows change mean CD/F1 by $(-3.58, +0.0163)$, $(-2.88, +0.0073)$, $(-2.61, +0.0134)$, and $(-0.11, +0.0006)$, respectively. These are descriptive changes under one fixed protocol rather than independent causal or statistical contribution estimates; the final alignment step is therefore a refinement of the complete configuration rather than the primary source of the gain.

Here $\hat{\epsilon}$ denotes the main denoiser prediction, ϵ_{weak} the auxiliary-head prediction, and s_{ig} the internal-guidance scale used in $\hat{\epsilon}_{\text{guided}} = \hat{\epsilon} + s_{\text{ig}}(\hat{\epsilon} - \epsilon_{\text{weak}})$. This notation is shared by the training and sampling descriptions.

The internal-guidance scale is a single fixed configuration value shared by the reported datasets and categories; no category-specific or dataset-specific scale search is used. Thus, the ablation establishes the behavior of the residual correction at $s_{\text{ig}} = 0.5$ and does not characterize the full scale-sensitivity curve. In particular, the present results do not imply that all positive scales are stable or that increasing the scale has a determined effect on output diversity.

B.12 Controlled Ablations on ShapeNet-R2N2 Chair

All controlled ablations use the deterministic 10% ShapeNet-R2N2 chair training subset and the same full 1,323-instance evaluable chair set. Each independently trained variant uses 10,000 optimization steps, a learning rate of 1.0×10^{-3}, batch size 16, and seed 42. Unless a row explicitly changes a factor, the model uses the full configuration summarized in Table 23. Sampling uses 1,000 DDPM steps and produces 4,096 points. CD and EMD are reported after multiplication by 10^3 and 10^2, respectively; EMD uses the exact 4,096-point protocol in Section E.2.

Table 23: Shared protocol for the controlled ShapeNet-R2N2 chair ablations.

Setting	Value
Training data	ShapeNet-R2N2 chair, deterministic 10% subset
Evaluation data	Full evaluable ShapeNet-R2N2 chair set (1,323 instances)
Optimization	10,000 steps, AdamW, LR 1.0×10^{-3}, batch size 16
Random seed	42
Output and sampling	4,096 points, DDPM, 1,000 denoising steps
Default serialization	Native FPS traversal order
Default recurrence	Chained forward/backward scan
Default retention	Log-spaced, token-conditioned decay with distance and time modulation
Visual conditioning	Frozen DINOv2 ViT-S/14 with registers; summed last four layers
Auxiliary objectives	Intermediate representation alignment and internal guidance enabled
Metrics	CD$\times 10^3$, F1 at $\tau = 0.01$, exact EMD@4096$\times 10^2$
Runtime profile	Batch size 1 at NFE $\{50, 100, 250, 500, 1000\}$; one timed repeat per NFE

Point-hierarchy scale ablation. We varied the number and allocation of point tokens at fixed effective training batch size on the deterministic 10% ShapeNet-R2N2 chair split. Single-scale variants use eight StateEvo blocks, while three-scale variants retain the $[2, 2, 4]$ block allocation. The three regimes pair approximately matched block-token budgets; the original three-scale model is included as the reference configuration and repeats the Full model row in Table 28. The high-budget three-scale run uses micro-batch 8 with two-step gradient accumulation, preserving an effective batch size of 16.

Across all three matched budgets, the three-scale layout improves CD, F1-score, and EMD relative to its single-scale counterpart. The advantage is largest at the low and middle budgets, while the high-budget comparison retains a smaller but consistent improvement across all three metrics.

Table 24: Point-hierarchy scale ablation on 10% ShapeNet-R2N2 chair. All models produce 4,096 points; CD and EMD are multiplied by 10^3 and 10^2, respectively.

Budget	Layout	Point counts	CD↓	F1↑	EMD↓
Low	Single-scale ([8])	[256]	69.61	0.4570	30.09
	Three-scale ([2, 2, 4])	[512, 256, 128]	**66.09**	**0.4720**	**29.50**
Middle	Single-scale ([8])	[512]	67.40	0.4680	29.46
	Three-scale (reference)	[1024, 512, 256]	**65.07**	**0.4800**	**29.02**
High	Single-scale ([8])	[1024]	67.01	0.4720	28.99
	Three-scale ([2, 2, 4])	[2048, 1024, 512]	**66.76**	**0.4780**	**28.66**

Table 25: Sequence-mixer block comparison on 10% ShapeNet-R2N2 chair. DSE denotes our StateEvo block; CD is multiplied by 10^3.

Block	CD↓	F1↑
Mamba	72.46	0.456
RWKV	80.01	0.431
DSE (Ours)	**64.96**	**0.481**

This is a block-replacement control within the same hierarchical denoiser: Mamba, RWKV, and StateEvo obtain CD/F1 values of 72.46/0.456, 80.01/0.431, and 64.96/0.481, respectively. The surrounding point grouping, visual conditioning, diffusion schedule, and evaluation protocol are held fixed. The comparison therefore reports the outcome of substituting the sequence-mixing block under this reconstruction setting; it is not a strictly parameter- or FLOP-matched estimate and does not imply that one mixer dominates across unrelated tasks or independently tuned model families.

Token serialization and distance modulation. Point tokens are evaluated under three traversal rules. Native FPS preserves the traversal inherited from hierarchical grouping, coordinate-sum sorting orders tokens by $x + y + z$, and Morton serialization uses a 10-bit-per-axis three-dimensional Z-order code after per-shape coordinate normalization. The native FPS implementation uses a random start point under the controlled experiment seed and is executed inside hierarchical grouping during each denoiser forward pass. Table 26 crosses these orders with distance-aware retention while keeping diffusion-time modulation enabled. Every row in this table is trained independently, so the comparison measures adaptation to each serialization as well as its interaction with geometric displacement.

Table 26: Token-serialization and distance-modulation ablation on 10% ShapeNet-R2N2 chair. Each row is trained independently; time modulation remains enabled.

Serialization	Dist.	Time	CD↓	F1↑	EMD↓
Native FPS traversal	On	On	**65.07**	**0.4800**	29.02
Coordinate-sum sorting	On	On	68.66	0.4710	29.52
Morton (Z-order)	On	On	67.06	0.4780	29.09
Native FPS traversal	Off	On	65.09	0.4790	**28.99**
Coordinate-sum sorting	Off	On	67.46	0.4720	29.29
Morton (Z-order)	Off	On	66.55	0.4780	29.05

Table 27: Same-checkpoint order-sensitivity analysis on 10% ShapeNet-R2N2 chair. All rows use the same model trained with native FPS traversal.

Sampling order	CD↓	F1↑	EMD@4096↓
Native FPS traversal	**65.07**	**0.4800**	**29.02**
Coordinate-sum sorting	76.90	0.4480	31.51
Morton (Z-order)	91.10	0.4030	34.29

The independently trained models show a moderate dependence on the traversal used during optimization: native FPS, coordinate-sum, and Morton ordering give CD/F1/EMD values of 65.07/0.4800/29.02, 68.66/0.4710/29.52, and 67.06/0.4780/29.09. A complementary same-checkpoint test changes only the sampling order and gives 65.07/0.4800/29.02 for native FPS, 76.90/0.4480/31.51 for coordinate-sum, and 91.10/0.4030/34.29 for Morton ordering. The paired results indicate that the recurrent denoiser adapts to its native traversal; serialization is therefore treated as part of the model configuration rather than as an interchangeable preprocessing step. The controlled seed provides reproducible runs, but does not make the random-start FPS rule invariant to its starting point.

Morton ordering is used here as a deterministic space-filling-curve control, not as evidence that all possible geometric traversals are equivalent. The reported protocol does not include training-time random serialization, a systematic sweep over FPS starting points, or an ensemble over multiple orders. Consequently, the measured differences describe the three specified deterministic orders and the same-checkpoint substitution test; they do not establish invariance to arbitrary order perturbations or provide an order-averaged model.

To isolate inference-time order sensitivity from retraining effects, Table 27 reuses one FPS-trained checkpoint and changes only the serialization used during sampling. The native FPS sampling row provides the same-checkpoint reference for measuring any change caused by the two alternative orders.

Decay parameterization and modulation. Table 28 separates the initialization of channel-wise memory scales, token-conditioned decay, adjacent-point distance modulation, and diffusion-time modulation. Log-spaced initialization places the per-head decay rates between 0.05 and 5.0 on a geometric grid, whereas free initialization samples them uniformly from $[0.5, 2.0]$. The effective distance and time coefficients are initialized to 1.0×10^{-3}. Token-invariant decay removes input-dependent decay logits but retains learned channel-wise decay; it is therefore a constant-across-token control rather than a no-decay model.

Table 28: Decay-component ablation on 10% ShapeNet-R2N2 chair. All variants retain the erase and write updates; only the listed decay factors change.

Variant	Init.	Token	Dist.	Time	CD↓	F1↑	EMD↓
Full model	Log-spaced	On	On	On	65.07	**0.4800**	29.02
Free decay initialization	Free	On	On	On	**64.63**	0.4780	28.85
Without distance modulation	Log-spaced	On	Off	On	65.09	0.4790	28.99
Without time modulation	Log-spaced	On	On	Off	65.12	0.4780	29.01
Without distance and time modulation	Log-spaced	On	Off	Off	65.81	0.4780	29.08
Free init., no distance/time modulation	Free	On	Off	Off	64.65	0.4780	**28.82**
Token-invariant decay	Free	Off	Off	Off	65.62	0.4770	28.91

The distance- and time-dependent terms produce small but complementary changes under this controlled setting. Relative to the full model, removing distance modulation changes CD/F1/EMD from 65.07/0.4800/29.02 to 65.09/0.4790/28.99, removing time modulation gives 65.12/0.4780/29.01, and removing both gives 65.81/0.4780/29.08. The modulation therefore acts as a retention control within the broader erase–write recurrence; it is not the sole source of the reconstruction gain.

The free-initialization control obtains slightly lower CD and EMD than the default log-spaced initialization in this single chair run. We treat this difference as an initialization-sensitivity observation rather than evidence that free initialization is generally preferable. The log-spaced default is retained because it explicitly spans multiple memory horizons at initialization and supplies a fixed, interpretable retention spectrum. More broadly, the small changes under single distance/time removals and the larger joint change are sensitivity diagnostics for conditional retention, not a claim that either modulation alone accounts for the full reconstruction gain. These controls use one category and one seed; they do not establish cross-category or seed-averaged robustness.

For the reported DDPM schedule, the training-time index is $t \in \{0, \dots, 1000\}$ and $T = 1000$. Reverse sampling visits these indices from high noise toward $t = 0$. With $\rho(t) = \exp(-\lambda_{\text{time}} t/T)$, larger t gives a smaller $\rho(t)$. Because the adapted log-decay is $g'_r = \rho(t)(g_r - |\boldsymbol{\lambda}_{\text{dist}}| \|p_r - p_{r-1}\|_2)$, this makes g'_r closer to zero and preserves more of the recurrent state at the noisy end of the trajectory. As t decreases toward zero, $\rho(t)$ approaches one and the distance-aware decay is applied more strongly, allowing the denoiser to emphasize local refinement near the final reconstruction. The reported ablation compares the complete time-modulation switch; it is not a measurement of a separate memory-length curve at every diffusion step.

Scan direction. The direction control in Table 29 distinguishes a single forward recurrence from two bidirectional constructions. Independent bidirectional averaging applies the forward and reversed recurrences to the same input values. Chained bidirectional scanning instead feeds the forward output to the reverse recurrence and averages the two outputs, allowing the second pass to refine information already propagated by the first.

Table 29: Scan-direction ablation on 10% ShapeNet-R2N2 chair.

Strategy	Passes	Reverse values	CD↓	F1↑	EMD↓
Forward only	1	Not applicable	**64.06**	**0.4830**	**28.70**
Independent bidirectional average	2	Original input	65.12	0.4810	28.91
Chained bidirectional average	2	Forward output	65.07	0.4800	29.02

The scan-direction comparison separates the effect of the recurrence layout from the effect of the retention parameterization. On this protocol, forward only gives the lowest CD and EMD and the highest F1 among the three variants, while the chained bidirectional scan remains the adopted configuration because its reverse pass refines the forward representation and exposes both traversal directions to the denoiser. Bidirectional processing is consequently treated as a contextual design choice rather than as a universal metric improvement. Because this comparison is restricted to one chair protocol and one seed, it is a contextual design diagnosis rather than evidence for cross-category or seed-averaged superiority of the adopted scan.

Frozen visual backbone. Table 30 evaluates sensitivity to the frozen representation teacher. Each backbone aggregates its last four transformer layers by summation, while surface projection, intermediate representation alignment, internal guidance, and the point-cloud denoiser remain enabled. Because the native feature widths and pretraining objectives differ, this experiment tests teacher dependence rather than constituting a parameter-matched architecture comparison.

Table 30: Frozen-visual-backbone sensitivity on 10% ShapeNet-R2N2 chair.

Frozen visual teacher	Patch	Dim.	Aggregation	CD↓	F1↑	EMD↓
DINOv2 ViT-S/14 with registers	14×14	384	Last 4, sum	**65.07**	**0.4800**	**29.02**
MSN ViT-S/16	16×16	384	Last 4, sum	71.87	0.4610	29.97
MAE ViT-B/16	16×16	768	Last 4, sum	68.01	0.4690	29.47

B.13 Exact EMD@4096 Comparison

Tables 31 and 32 report category-wise mean EMD under the exact 4,096-point assignment protocol in Section E.2. Every displayed value is multiplied by 10^2, and lower values indicate better global point-set correspondence. PDM is not included in this metric-specific comparison because matching exact EMD@4096 results are unavailable; CD and F1 comparisons with PDM are reported in the main paper. Best and second-best available values are highlighted.

Table 31: Exact EMD@4096 on the main ShapeNet-R2N2 categories.

Category	Training data	BDM-B	MESC-3D	RepStateDiff
Airplane	1%	39.20	92.34	**36.92**
	10%	35.65	93.22	**29.14**
	50%	32.53	95.09	**25.33**
	100%	31.55	88.35	**24.17**
Car	1%	28.17	66.10	**28.03**
	10%	26.29	50.51	**24.80**
	50%	25.71	49.37	**23.32**
	100%	25.60	48.09	**23.28**
Chair	1%	**34.61**	74.28	35.21
	10%	33.41	78.06	**28.97**
	50%	37.34	79.30	**25.82**
	100%	27.58	82.56	**24.63**

Table 32: Exact EMD@4096 on the remaining ShapeNet-R2N2 categories at 10% training data.

Category	BDM-B	MESC-3D	RepStateDiff
Bench	28.28	69.33	**25.85**
Cabinet	31.63	86.36	**26.52**
Display	31.82	84.79	**29.17**
Lamp	50.56	115.84	**41.11**
Loudspeaker	33.67	72.72	**32.11**
Rifle	28.15	94.41	**26.78**
Sofa	27.72	86.38	**25.99**
Table	35.79	89.83	**27.34**
Telephone	22.33	70.36	**22.09**
Watercraft	28.40	92.42	**26.65**

Across the reported ShapeNet-R2N2 settings, RepStateDiff obtains the lowest EMD in all but the 1% chair setting. These results complement CD and F1-score by measuring global one-to-one point correspondence.

C THEORETICAL ANALYSIS OF STATEEVO

This section records algebraic properties of the StateEvo recurrence used in the denoiser. All statements are conditional on the token projections and gates at a given denoising call; they describe the recurrent operator itself and do not constitute task-level accuracy, convergence, or robustness claims.

C.1 LOCAL OBJECTIVE INTERPRETATION

For one token index r, define the channel-wise decay matrix $D_r = \mathrm{diag}(\gamma_r)$, the decayed state $\bar{S}_r = D_r S_{r-1}$, the erase direction $e_r = \alpha_r \odot k_r$, and the gated write target $z_r = w_r \odot v_r$. The recurrence can then be written as

$$S_r = \bar{S}_r + k_r \big(z_r - \bar{S}_r^\top e_r\big)^\top. \tag{14}$$

Proposition 1 (exact local objective). For fixed $\bar{S}_r$, k_r, e_r, and z_r, the update in Eq. 14 is the unique minimizer of

$$\mathcal{L}_r(S) = \frac{1}{2}\|S - \bar{S}_r\|_F^2 - \big\langle S^\top k_r,\ z_r - \bar{S}_r^\top e_r \big\rangle. \tag{15}$$

Proof. Let $c_r = z_r - \bar{S}_r^\top e_r$. The gradient and Hessian with respect to S are

$$\nabla_S \mathcal{L}_r(S) = S - \bar{S}_r - k_r c_r^\top, \qquad \nabla_S^2 \mathcal{L}_r(S) = I. \tag{16}$$

The Hessian is positive definite, so the objective is strictly convex. Setting the gradient to zero gives $S = \bar{S}_r + k_r c_r^\top$, which is exactly Eq. 14. Therefore the minimizer is unique. □

The first term in Eq. 15 keeps the new state close to the decayed memory, while the linear term applies an associative correction toward the gated value target after reading the old content along the gated erase direction. The result is a per-token characterization of the implemented update; it is not a claim that the complete diffusion objective is convex.

C.2 RETENTION RANGE AND ONE-STEP STATE BOUND

The decay parameterization also gives a direct bound on the state before the erase–write correction. Assume $g_r \le 0$, the learned distance coefficient is non-negative, and the time-decay gain is non-negative, as in the reported configuration. Let $\Delta p_r = \|p_r - p_{r-1}\|_2$ and $\rho(t) = \exp(-\lambda_{\mathrm{time}} t/T)$. Then

$$g'_r = \rho(t)\big(g_r - |\boldsymbol{\lambda}_{\mathrm{dist}}|\Delta p_r\big) \le 0, \qquad 0 < \gamma'_r = \exp(g'_r) \le 1. \tag{17}$$

In the reported implementation, A_d denotes a positive channel-wise decay rate and the trainable log-parameter is $a_d = \log A_d$; consequently, $\exp(a_d) = A_d$ is the factor used in the log-decay gate. The log-spaced initialization therefore changes the initial decay spectrum without introducing an additional recurrent state. For distance modulation, the serialized coordinate difference is assigned to the token that has a predecessor in the corresponding scan: the first forward token has zero distance, and the first token of the reversed traversal (the last token in the original order) also has zero distance. The remaining distances are reversed together with the scan.

Lemma 1 (decay cannot amplify the incoming state). Under the conditions above,

$$\|\mathrm{diag}(\gamma'_r) S_{r-1}\|_F \le \|S_{r-1}\|_F. \tag{18}$$

Proof. For a diagonal matrix, left multiplication scales each row of S_{r-1} by the corresponding coordinate of γ'_r. Equation 17 gives $0 < \gamma'_{r,j} \le 1$ for every key channel j. Hence

$$\|\mathrm{diag}(\gamma'_r) S_{r-1}\|_F^2 = \sum_j (\gamma'_{r,j})^2 \|(S_{r-1})_{j,:}\|_2^2 \le \sum_j \|(S_{r-1})_{j,:}\|_2^2. \tag{19}$$

Taking square roots proves Eq. 18. □

Moreover, $\rho(t)$ is non-increasing in t for $\lambda_{\mathrm{time}} \ge 0$. Thus larger reverse-process indices attenuate the magnitude of the negative log-decay and move the retention factor toward one, whereas $t \to 0$ recovers the full distance-aware log-decay. The bound applies to the decay operation only; the subsequent rank-one edit can increase the full state norm and is not claimed to be globally non-expansive.

C.3 Exact Chunkwise Form and Linear Sequence Complexity

The recurrence admits an exact chunkwise representation. Consider one scan and a chunk of C tokens, with S_0 denoting the state at the chunk entrance. Define the cumulative decay vectors

$$\Gamma_r = \bigodot_{i=1}^{r} \gamma_i, \qquad \Gamma_0 = \mathbf{1}, \qquad S_r = \mathrm{diag}(\Gamma_r)\widehat{S}_r, \tag{20}$$

and the normalized factors

$$\bar{k}_r = \Gamma_r^{-1} \odot k_r, \qquad \bar{e}_r = \Gamma_r \odot e_r. \tag{21}$$

Because every component of Γ_r is positive, the inverse in Eq. 21 is well-defined.

Theorem 1 (exact chunkwise equivalence). Let $\bar{K}$, $\bar{E}$, and Z stack the rows $\bar{k}_r^\top$, $\bar{e}_r^\top$, and $z_r^\top$, respectively. Define

$$T = \mathrm{tril}(\bar{E}\bar{K}^\top, -1), \qquad A = (I+T)^{-1}, \qquad Y = A\bar{E}, \qquad U = AZ. \tag{22}$$

Then the normalized prefix state is

$$\widehat{S}_r = S_0 + \bar{K}_{1:r}^\top (U_{1:r} - Y_{1:r} S_0), \tag{23}$$

and the end-of-chunk state is

$$S_C = \mathrm{diag}(\Gamma_C) S_0 + K_{\mathrm{tail}}^\top (U - Y S_0), \tag{24}$$

where row r of K_{tail} is $(\Gamma_C \oslash \Gamma_r) \odot k_r$.

Proof. Substituting Eq. 20 into the recurrence and multiplying by $\mathrm{diag}(\Gamma_r)^{-1}$ gives

$$\widehat{S}_r = (I - \bar{k}_r \bar{e}_r^\top)\widehat{S}_{r-1} + \bar{k}_r z_r^\top. \tag{25}$$

Writing the rank-one increment as $\bar{k}_r d_r^\top$, Eq. 25 gives

$$d_r^\top = z_r^\top - \bar{e}_r^\top \widehat{S}_{r-1}. \tag{26}$$

Induction on the prefix length yields $\widehat{S}_{r-1} = S_0 + \sum_{s<r} \bar{k}_s d_s^\top$ and therefore

$$d_r^\top = z_r^\top - \bar{e}_r^\top S_0 - \sum_{s<r} (\bar{e}_r^\top \bar{k}_s) d_s^\top. \tag{27}$$

Stacking these equations over the chunk gives $(I+T)D = Z - \bar{E}S_0$, where D contains the rows $d_r^\top$. Since T is strictly lower triangular, $I+T$ is unit lower triangular and invertible. Thus $D = U - YS_0$, which proves Eq. 23. Multiplying the prefix formula at $r = C$ by $\mathrm{diag}(\Gamma_C)$ gives Eq. 24. □

For fixed chunk size C and fixed feature dimensions, forming T, solving the unit lower-triangular system, and applying the state update costs

$$O\left(\frac{L}{C}\left[C^2(D_k + D_v) + C D_k D_v\right]\right) = O(L) \quad \text{in the sequence length } L. \tag{28}$$

The computation uses matrices whose first dimension is the fixed chunk size, not the full sequence length, and therefore does not construct an $L \times L$ attention matrix. The default chained bidirectional readout performs a fixed number of such scans, so its asymptotic dependence on L remains linear; the scan direction changes constants rather than the order.

C.4 Tied-Gate Reduction

The channel-wise recurrence contains the scalar-gated update as a special case. If $\alpha_r = \beta_r \mathbf{1}_{D_k}$ and $w_r = \beta_r \mathbf{1}_{D_v}$, then $e_r = \beta_r k_r$, $z_r = \beta_r v_r$, and Eq. 14 reduces to

$$S_r = (I - \beta_r k_r k_r^\top) D_r S_{r-1} + \beta_r k_r v_r^\top. \tag{29}$$

If the decay is also tied across key channels, $D_r = \gamma_r I$, this becomes the scalar-decay form. The reduction follows by direct substitution, so the channel-wise formulation strictly includes the tied-gate recurrence while allowing the erase and write axes to use different channel patterns.

These results are algebraic properties of the recurrent operator. They do not imply convergence of the complete diffusion sampler, reconstruction accuracy, invariance to token order, robustness to camera error, or superiority over other sequence mixers; each of those claims requires separate empirical or task-specific theoretical evidence.

Table 33: Mechanistic positioning of StateEvo relative to representative linear-time sequence mixers. The table distinguishes recurrent primitives from their use in image-conditioned point-cloud diffusion; it does not imply an empirical ranking among the sequence mixers.

Method	Recurrent state	Retention or update	Selective overwrite	Geometry and diffusion use
Linear attention Katharopoulos et al. (2020)	Key–value accumulator	Additive outer-product updates; no inherent forgetting	No explicit key-directed erase	Generic sequence mixing
RetNet Sun et al. (2023)	Decayed key–value accumulator	Prescribed multi-scale exponential retention	No explicit erase/write separation	Generic parallel, recurrent, or chunkwise sequence modeling
Mamba Gu & Dao (2024)	Selective SSM states	Input-dependent discretization through selective state-space parameters	Content selection is expressed through SSM parameters rather than key-directed memory editing	Generic causal sequence scan
RWKV / PointRWKV Peng et al. (2023); He et al. (2025)	Time-mixed key–value state; PointRWKV uses hierarchical point states	Learned time decay with receptance-gated readout	No independent key-side erase and value-side write in the RWKV update	PointRWKV targets representation learning from observed point clouds
StateEvo	Key–value matrix state	Channel-wise decay, erase, and write, further modulated by adjacent 3D distance and diffusion time	Separate key-directed erase and value write	Chained forward/backward scans over surface-aligned, image-conditioned hierarchical point tokens

D Comparison with Related Methods

D.1 Relationship to Linear Attention and State-Space Models

StateEvo is positioned within efficient recurrent sequence mixing rather than as a new state-space primitive. Kernelized linear attention accumulates key–value outer products in a fixed-size recurrent state Katharopoulos et al. (2020). RetNet introduces exponentially decayed retention and equivalent parallel, recurrent, and chunkwise-recurrent forms Sun et al. (2023). RWKV combines time-decayed key–value aggregation with a receptance gate Peng et al. (2023), whereas Mamba makes the discretized state-space parameters input selective Gu & Dao (2024). These mechanisms motivate linear-complexity alternatives to dense attention; unlike their generic sequence-modeling formulations, StateEvo is embedded in an image-conditioned point-cloud denoiser whose retention is modulated by serialized 3D displacement and diffusion time.

Scope of the linear-complexity statement. The linear-complexity statement for StateEvo applies to the token scan with fixed channel dimensions. For L serialized tokens, each scan updates a fixed-size $D_k \times D_v$ recurrent state, so its computational cost grows linearly with L and does not require pairwise token interactions. This statement concerns the sequence-mixing block only; it does not imply linear end-to-end complexity for the complete reconstruction pipeline, which also includes image encoding, camera projection, hierarchical grouping, local refinement, and repeated reverse-diffusion evaluations.

RetNet. RetNet replaces multi-head attention with gated multi-scale retention, assigning each head a fixed exponential decay and providing equivalent parallel, recurrent, and chunkwise-recurrent forms for autoregressive language modeling Sun et al. (2023). Its recurrent state follows a decayed key–value accumulation without an explicit erase/write separation, while its swish gate modulates the retention output. StateEvo instead learns channel-wise decay within a point-cloud denoiser, uses separate key-directed erase and value-side write gates, and further modulates retention by adjacent 3D displacement and diffusion time. It also applies chained forward/backward scans to image-conditioned point tokens and does not cache the matrix state across reverse-diffusion steps. The distinction therefore lies in geometry- and denoising-conditioned state editing, rather than in exponential retention itself.

Erase, write, and exponential decay are established recurrent primitives. StateEvo specializes and couples them for point-cloud diffusion: retention is conditioned on displacement between adjacent serialized 3D tokens and on the diffusion timestep; the recurrent mixer operates on hierarchical

tokens that fuse noisy geometry with camera-projected image evidence; and the backward scan consumes the forward output to refine context from the opposite direction. The matrix state is initialized and propagated across serialized tokens within each denoiser call rather than cached across reverse-diffusion steps. Across diffusion steps, persistence is carried by the evolving noisy point hypothesis.

PointRWKV is a particularly relevant 3D comparator because it already combines hierarchical point patches, efficient recurrent global mixing, bidirectional processing, and a local graph stabilizer He et al. (2025). StateEvo is therefore not distinguished by point serialization or bidirectionality alone. PointRWKV learns representations of an observed point cloud for analysis tasks such as classification and segmentation. StateEvo instead mixes noisy, image-conditioned point tokens inside a conditional diffusion denoiser and explicitly adapts retention to serialized 3D displacement and diffusion time. Accordingly, the contribution is the geometry- and denoising-aware integration of these operations, rather than a general claim of inventing recurrent sequence modeling.

Point Cloud Mamba and Pamba further demonstrate that point-cloud sequence construction is an active design space: Point Cloud Mamba uses consistent traverse serialization, whereas Pamba combines multi-path serialization with a ConvMamba block Zhang et al. (2025); Li et al. (2025b). These methods target point-cloud analysis and segmentation, while StateEvo is used inside an image-conditioned diffusion denoiser. Thus, the distinction is the denoising- and geometry-conditioned erase/write/retention update, not a claim that StateEvo is the only method to address serialization discontinuities.

The erase, write, and exponential-decay operations used by StateEvo are established recurrent primitives, and the formulation does not by itself imply a new state-space expressiveness theorem. The evidence provided here supports the mechanistic distinction between a generic mixer and a geometry- and denoising-conditioned mixer, together with the controlled block substitution in Table 25. It does not establish a universal expressiveness ordering or a cross-task theoretical superiority claim over linear attention, RWKV, Mamba, or other state-space models.

D.2 Relationship to Representation Alignment and Distillation

The intermediate alignment loss is explicitly related to the broader feature-distillation and representation-alignment lineage Romero et al. (2015); Yu et al. (2025). In this broad sense, frozen DINOv2 acts as a representation teacher: the denoiser learns point features that approach a fixed visual target while the teacher itself is not updated. We use this terminology only for representation supervision, rather than for teacher–student model compression. RepStateDiff does not copy task logits, compress a pretrained denoiser, or match a teacher's reverse trajectory.

REPA aligns hidden states of a diffusion transformer with representations produced by an external pretrained visual encoder Yu et al. (2025). RepStateDiff follows the same high-level principle of supervising an internal denoising representation with frozen visual features, but changes the correspondence being learned. Instead of assigning an image-level or patch-token target in the generative image domain, it samples DINOv2 features from the observed conditioning image at camera-projected noisy 3D points and aligns them with decoded point features at the same diffusion timestep. The target is therefore cross-modal, view-conditioned, and point indexed. The shared idea is representation-level supervision, not an assertion that the two methods use the same objective, representation domain, or generative role.

SRA is also related at the level of representation-guided denoising: it obtains guidance from the diffusion transformer itself by aligning an earlier-layer, higher-noise representation with a later-layer, lower-noise target, without an external representation encoder Jiang et al. (2025). RepStateDiff instead uses an external frozen DINOv2 target and does not construct a dual-time self-distillation target. Its weak noise head is not a target network or teacher for the alignment loss. It shares the denoising trunk, is jointly supervised to predict the injected noise, and supplies a sampling correction derived from the same decoded point features. RepStateDiff also differs from diffusion sampler distillation Salimans & Ho (2022): there is no pretrained diffusion teacher, trajectory matching, student compression, or reduction in the number of denoising evaluations.

Scope of the representation-guidance claim. Feature alignment, self-guidance, and distillation are established forms of representation supervision. In RepStateDiff, their role is geometry-specific: camera projection converts frozen image features into targets indexed by the current noisy 3D hy-

pothesis, and a jointly trained auxiliary head modifies noise prediction from the same point-state representation. The resulting contribution is cross-modal, point-indexed representation guidance coupled to state-evolution denoising.

Dependence on the frozen visual teacher. Representation guidance can inherit the inductive biases of DINOv2, including sensitivity to its pretraining distribution and preference for semantic or appearance cues that need not map perfectly to occluded 3D geometry. Freezing the encoder stabilizes the target and limits trainable parameters, but does not remove these biases. The MSN/MAE/DINOv2 comparison therefore serves as a teacher-dependence diagnostic under the same 10% ShapeNet-R2N2 chair protocol as the other controlled ablations.

E PRELIMINARY

We summarize the conditional point-cloud diffusion formulation and the evaluation metrics used throughout the supplementary material. The notation follows the main paper. For compactness, the image and camera condition is denoted by c.

E.1 CONDITIONAL POINT CLOUD DIFFUSION MODEL

Diffusion models generate point clouds by learning the reverse of a gradual noising process. Let $X_0 \in \mathbb{R}^{N\times 3}$ be a clean point cloud sampled from the data distribution p_{data}, where N is the number of points. The forward process progressively corrupts X_0 into X_T through a Markov chain:

$$q(X_{1:T} \mid X_0) = \prod_{t=1}^{T} q(X_t \mid X_{t-1}), \tag{30}$$

where each transition is Gaussian:

$$q(X_t \mid X_{t-1}) = \mathcal{N}\Big(X_t; \sqrt{1-\beta_t}X_{t-1}, \beta_t \mathbf{I}\Big). \tag{31}$$

Here $\{\beta_t\}_{t=1}^{T}$ is a predefined variance schedule and $\mathbf{I}$ denotes the identity covariance. With $\alpha_t = 1-\beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$, the marginal distribution at any timestep has the closed form

$$q(X_t \mid X_0) = \mathcal{N}\big(X_t; \sqrt{\bar{\alpha}_t}X_0, (1-\bar{\alpha}_t)\mathbf{I}\big), \tag{32}$$

or equivalently

$$X_t = \sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{33}$$

As t increases, $\bar{\alpha}_t$ decreases and the point cloud approaches an isotropic Gaussian distribution.

The reverse process recovers the clean point cloud from noise. In image-conditioned reconstruction, the denoising transition depends on both the noisy point cloud X_t and the external condition c:

$$p_\theta(X_{0:T} \mid c) = p(X_T)\prod_{t=1}^{T} p_\theta(X_{t-1} \mid X_t, c), \tag{34}$$

where

$$p_\theta(X_{t-1} \mid X_t, c) = \mathcal{N}\big(X_{t-1}; \mu_\theta(X_t, t, c), \sigma_t^2\mathbf{I}\big). \tag{35}$$

The network-predicted mean $\mu_\theta(X_t, t, c)$ parameterizes the denoising direction, while σ_t^2 controls the variance at each reverse step.

Training can be derived from the variational lower bound:

$$\begin{aligned}\mathcal{L}_{\text{VLB}} = \mathbb{E}_q[-\log p_\theta(X_0 \mid X_1, c)] \\ + D_{\text{KL}}(q(X_T \mid X_0)\|p(X_T)) \\ + \sum_{t=2}^{T} D_{\text{KL}}(q(X_{t-1} \mid X_t, X_0) \\ \|p_\theta(X_{t-1} \mid X_t, c)).\end{aligned} \tag{36}$$

The first term reconstructs the clean point cloud near $t = 0$, the second term regularizes the terminal distribution, and the remaining KL terms match the learned reverse transitions to the true diffusion posterior.

In practice, the VLB is commonly optimized through the simplified noise-prediction loss:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,X_0,c,\epsilon}\Big[\big\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t, c)\big\|_2^2\Big], \tag{37}$$

where $\epsilon_\theta(\cdot)$ predicts the Gaussian noise injected into X_0. This objective encourages the denoiser to remove time-dependent noise while using c to keep the reconstructed point cloud consistent with the input image.

E.2 Quantitative Metrics

We evaluate reconstruction quality with Chamfer Distance (CD), F1-score, and Earth Mover's Distance (EMD). ShapeNet-R2N2 coordinates are standardized with the split-level point-cloud mean and standard deviation used by its loader, whereas Pix3D shapes are centered and divided by their scalar coordinate standard deviation; the reported scale factor is 1.0 for both datasets. Before computing these metrics, the prediction and reference are independently centroid aligned; no rotational, scale, or ICP alignment is applied. For a predicted point set $\mathcal{P}$ and ground-truth point set $\mathcal{G}$, CD measures the bidirectional nearest-neighbor discrepancy:

$$\begin{aligned}\mathrm{CD}(\mathcal{P},\mathcal{G}) = &\frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\min_{g\in\mathcal{G}}\|p-g\|_2^2 \\ &+ \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}}\min_{p\in\mathcal{P}}\|g-p\|_2^2.\end{aligned} \tag{38}$$

We report CD multiplied by 10^3 in all tables.

Since CD can be sensitive to outliers, we also report F1-score at threshold $\tau = 0.01$. Consistent with the released evaluator, the threshold is applied to squared Euclidean nearest-neighbor distances. Precision and recall are defined as

$$\begin{aligned}\mathrm{Prec}_\tau &= \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\mathbf{1}\left[\min_{g\in\mathcal{G}}\|p-g\|_2^2 < \tau\right], \\ \mathrm{Rec}_\tau &= \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}}\mathbf{1}\left[\min_{p\in\mathcal{P}}\|g-p\|_2^2 < \tau\right].\end{aligned} \tag{39}$$

The F1-score is their harmonic mean:

$$\mathrm{F1}_\tau = \frac{2\,\mathrm{Prec}_\tau\,\mathrm{Rec}_\tau}{\mathrm{Prec}_\tau + \mathrm{Rec}_\tau}, \quad \tau = 0.01. \tag{40}$$

EMD measures a global one-to-one correspondence rather than independent nearest neighbors. For two equally sized point sets of N points, it is

$$\mathrm{EMD}(\mathcal{P},\mathcal{G}) = \min_{\pi\in\mathfrak{S}_N}\frac{1}{N}\sum_{i=1}^{N}\left\|p_i - g_{\pi(i)}\right\|_2, \tag{41}$$

where $\mathfrak{S}_N$ is the set of permutations of N elements. We report **EMD@4096**: after independently centering the complete 4096-point prediction and reference, we construct the full pairwise Euclidean cost matrix and solve the resulting linear-sum assignment exactly. No point subsampling is applied. Every method uses the same file-matching, centering, and assignment protocol. We report EMD $\times$ 10^2; lower is better.

Lower CD and EMD and higher F1-score indicate better reconstruction quality. Independent centroid alignment follows the existing CD/F1 implementation but removes global translation error; this convention must therefore be held fixed when comparing methods. These protocol-specific distances quantify geometric proximity under the stated normalization and sampling resolution; they are not direct measures of visual fidelity, manufacturability, or downstream application utility.

F Implementation Details

This section reports the RasterProj implementation together with the computing infrastructure and training hyperparameters used by the RepStateDiff experiments. All runs use the same workstation configuration and the same model architecture; only the training-data ratio, learning rate, and number of optimization steps vary across the low-data settings.

Computing infrastructure. Experiments were run on Ubuntu 24.04 with an Intel Core i7-14700KF CPU, one NVIDIA RTX 3090 GPU, and 32 GB system memory. The tested environment uses Python 3.10.20, PyTorch 2.6.0+cu124, torchvision 0.21.0+cu124, torchaudio 2.6.0+cu124, CUDA toolkit/compiler/NVCC 12.9, and GCC/G++ 13.4.0. The main Python dependencies are taken from the Weights & Biases exported runtime requirements and the submitted code environment: PyTorch3D 0.7.8 (installed from commit `75ebeea` in the code archive), diffusers 0.38.0, accelerate 1.13.0, transformers 5.8.1, timm 1.0.27, Open3D 0.19.0, Trimesh 4.12.2, OpenCV 4.11.0.86, numpy 1.26.4, scipy 1.15.3, scikit-learn 1.7.2, pandas 2.3.3, Hydra 1.3.2, OmegaConf 2.3.0, and Weights & Biases 0.27.0. The training logs record PyTorch Hub model loading, a Hugging Face Transformers cosine scheduler, Weights & Biases logging, and a frozen DINOv2 ViT-S/14 image encoder with registers.

Table 34: Computing infrastructure and software components.

Item	Setting
Operating system	Ubuntu 24.04
CPU	Intel Core i7-14700KF
GPU	1× NVIDIA RTX 3090
System memory	32 GB
Implementation	Python 3.10.20, PyTorch 2.6.0+cu124
CUDA and compilers	CUDA toolkit/NVCC 12.9, GCC/G++ 13.4.0
3D/CV libraries	PyTorch3D 0.7.8 (`75ebeea`), Open3D 0.19.0, Trimesh 4.12.2, OpenCV 4.11.0.86
Training libraries	diffusers 0.38.0, accelerate 1.13.0, transformers 5.8.1, timm 1.0.27
Scientific stack	numpy 1.26.4, scipy 1.15.3, scikit-learn 1.7.2, pandas 2.3.3
Logging/config	wandb 0.27.0, Hydra 1.3.2, OmegaConf 2.3.0
Sequence kernels	fla-core 0.5.0, flash-linear-attention 0.5.0
Feature encoder loading	PyTorch Hub, DINOv2 ViT-S/14 with registers
xFormers	Not enabled in the logged runs

Shared training configuration. The final model uses a DDPM denoising scheduler with a linear beta schedule. Images are resized to 224×224, and each supervised shape contains 4096 points. We train with AdamW, cosine learning-rate decay, and a fixed random seed. Table 35 lists the shared settings extracted from the training logs.

Table 35: Shared implementation and optimization hyperparameters.

Hyperparameter	Value
Optimizer	AdamW
AdamW betas	$(0.95, 0.999)$
Weight decay	1.0×10^{-6}
Batch size per device	16
Gradient accumulation	1
Gradient clipping	50
Scheduler	Cosine
Warmup steps	2,000
Mixed precision	No
EMA	Disabled
Input image size	224×224
Supervised points	4096
Data-loader workers	8
DDPM beta schedule	Linear, 1.0×10^{-5} to 0.008
Intermediate alignment weight	0.1
Weak-head loss weight	0.1
Internal guidance scale	$s_{\mathrm{ig}} = 0.5$
Total / trainable parameters	38.11M / 16.05M
Peak logged training memory	About 20.7 GB

Training ratios. Table 36 summarizes the settings used for the ShapeNet-R2N2 chair runs in the provided logs. The number of DDPM inference steps is fixed to 1000 for all training-data ratios, while the number of optimization steps increases with the amount of available training data.

Table 36: Hyperparameters for different ShapeNet-R2N2 chair training ratios.

Ratio	Train shapes	Batches/epoch	Train steps	LR	DDPM steps
1%	52	3	4,000	1.0×10^{-3}	1,000
10%	526	32	10,000	1.0×10^{-3}	1,000
25%	1,321	82	25,000	8.0×10^{-4}	1,000
50%	2,640	165	50,000	5.0×10^{-4}	1,000
100%	5,265	329	100,000	3.0×10^{-4}	1,000

F.1 RasterProj Implementation Details

In the reported configuration, the frozen DINOv2 ViT-S/14 encoder with registers returns normalized patch tokens from its last four transformer layers. These tokens are summed, reshaped into a spatial feature grid, and bilinearly upsampled to the 224×224 input resolution. The local conditioning map concatenates normalized RGB with the 384-dimensional visual feature map. Foreground-mask, distance-transform, and global class-token channels are disabled, giving 387 image-conditioning channels and 390 channels after the noisy 3D coordinates are concatenated. For point i at timestep t, the resulting denoiser input can be written as $u_i^t = [x_i^t; r_i^t; c_i^t]$, where r_i^t and c_i^t are the projected RGB and visual feature vectors, respectively.

Given $X_t \in \mathbb{R}^{B \times N \times 3}$, RasterProj first verifies that the conditioning map has the same spatial resolution as the input image. It clones the PyTorch3D camera and scales its translation by the dataset scale factor, which is 1.0 in the reported experiments. A `PointsRasterizer` is then configured with image size 224×224, point radius 0.0075, one point per pixel, and `bin_size=0`. Rasterization returns a point-index tensor of shape $B \times 224 \times 224 \times 1$; entries greater than or equal to zero identify valid point–pixel assignments. The feature at each valid pixel is written to the corresponding row of a zero-initialized $B \times N \times 387$ point-feature buffer. Points without a valid rasterized assignment therefore retain a zero image condition. When cameras are supplied as a list, the same operation is applied sample by sample and the projected outputs are concatenated across the batch.

The one-point-per-pixel setting also fixes the collision rule. If several points fall within the same pixel footprint, PyTorch3D returns the single depth-ordered index retained by the rasterizer; the implementation does not average, pool, or learn an additional feature aggregation over those points. Points projected outside the image, occluded by another point at a pixel, or otherwise absent from the returned index tensor retain the zero-initialized image feature for that projection. Thus, the projection is a sparse point–pixel lookup with depth selection, rather than a dense visibility completion operation.

During sampling, the RGB and DINOv2 conditioning map is computed once and cached, but RasterProj is called again inside every reverse-diffusion iteration using the current X_t. Thus, the image representation remains fixed while point–pixel assignments are updated as the noisy coordinates evolve. The projected condition is concatenated with X_t to form the $B \times N \times 390$ denoiser input. Training follows the same projection path for the randomly noised X_t, and the projected DINOv2 features are also reused as the point-indexed target of intermediate representation alignment. The normalization in the alignment loss is along the final feature/channel dimension. Zero-filled projected targets identify invalid assignments, and only points with a nonzero projected target enter the alignment loss.

Each reverse step first obtains the cached image features and projects them at the current X_t, then builds the hierarchical tokens and applies the StateEvo scan. The decoded features pass through the local detail refiner and the weak noise head, after which internal guidance produces the final noise prediction used by the DDPM scheduler to update X_t to X_{t-1}. The next step repeats the projection with the new point hypothesis; no recurrent matrix state is carried between diffusion steps.

At test time, the camera condition C is constructed from the view and camera metadata associated with each benchmark image and passed directly to RasterProj. It is not inferred from the RGB image, recovered from the target point cloud, or optimized during sampling.

The projection assumes that the dataset-provided camera is available and accurate, including the intrinsics, extrinsics, and the convention that the camera is scaled consistently with the normalized point coordinates. The implementation does not estimate missing camera parameters, inject camera noise during sampling, or optimize the pose jointly with the denoising trajectory. Robustness to camera perturbations and unknown-camera inference is therefore outside the evidence of the present experiments; pose estimation or sampling-time camera optimization would be an extension of the conditioning interface rather than a property evaluated here.

G Limitations and Broader Impact

Limitations. RepStateDiff is evaluated under established ShapeNet-R2N2, Pix3D, CO3Dv2, and Objaverse single-view point-cloud reconstruction protocols. The primary comparisons use category-specific models and the cross-dataset results are reported as diagnostics rather than a single merged ranking. Future work can examine unseen-category transfer, accelerated samplers for lower-latency reconstruction, and stochastic output coverage. The current evaluation reports CD, F1-score, EMD, and efficiency diagnostics; it does not include COV, MMD, or a fidelity–diversity curve. The reported results should therefore be read as quality and efficiency evidence, without a claim about multimodal coverage or sample diversity under ambiguous images.

Broader Impact. Single-view 3D reconstruction can support robotics, AR/VR, object digitization, and interactive 3D content creation by reducing the amount of geometry supervision required for model training. Potential negative uses include reconstructing objects or scenes without appropriate consent, producing deceptive digital assets, or integrating reconstruction into surveillance-oriented pipelines. The present work is evaluated on public object-centric benchmarks and does not release a human-centered dataset; responsible deployment should still respect data ownership, privacy, and downstream safety constraints. Because the conditioning backbone is frozen, its pretraining distribution can also transfer appearance and semantic biases into the reconstructed geometry. Deployment should therefore document data and camera provenance and audit performance across relevant object categories and visual conditions before use in consequential settings.

H Visualization Results

This section collects the qualitative comparisons so that the numerical results and visual evidence remain easy to inspect separately. The figures use the same reconstructions and training-data regimes described in the main paper and the preceding experimental section. In addition to the synthetic ShapeNet-R2N2 and real-image Pix3D examples, we include representative CO3Dv2 reconstructions drawn from the 12-category subset shown in the full-data evaluation of Table 15.

For ShapeNet-R2N2, the cross-method panel covers the regimes with a common set of displayed baseline outputs, while the 1% regime is represented by the complete quantitative results in Table 1. The Pix3D panel additionally includes the 1%, 10%, and 100% regimes.

The CO3Dv2 examples in Figure 7 complement the category-level metrics with a direct visual check on real images. Each triplet contains the input image, the RepStateDiff reconstruction, and the ground-truth point cloud. The examples span varied object appearances, viewpoints, and geometric structures; the predictions generally preserve the coarse silhouette and major parts, while the remaining discrepancies are most visible around thin or partially observed regions. This figure is a qualitative supplement to Table 15, rather than a cross-method ranking.

The local detail refiner is intended to improve point-level consistency on visible surfaces using image-aligned features. Thin, occluded, and weakly observed structures remain intrinsically ambiguous, and residual discrepancies can persist around wheels, chair backrests, and other narrow parts. The visual examples should therefore be read as evidence of local surface consistency under image conditioning, alongside rather than in place of a dedicated fine-structure evaluation.

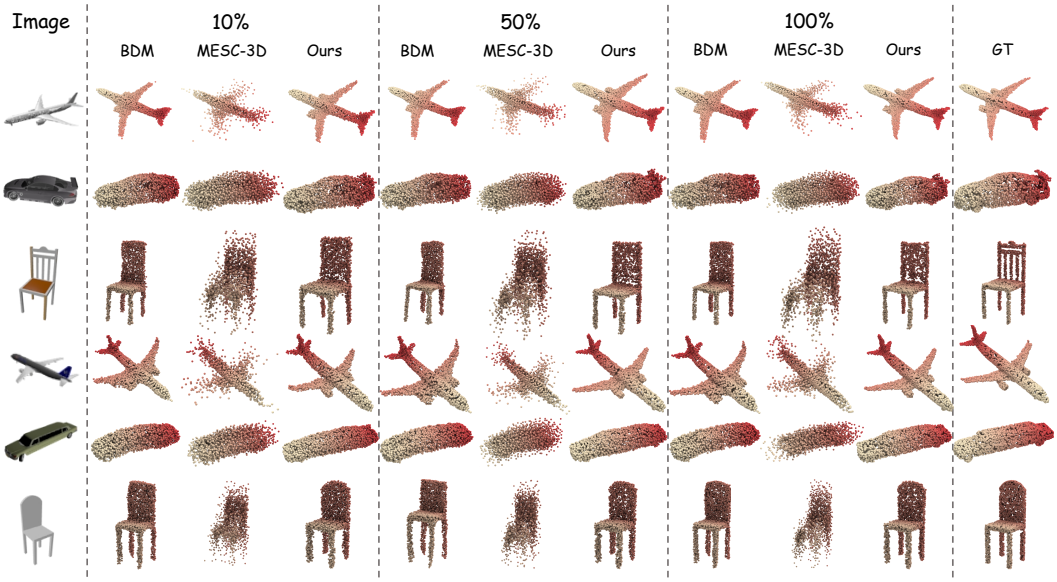

Figure 5: Qualitative ShapeNet-R2N2 comparison across 10%, 50%, and 100% training-data regimes. For each regime, the available BDM, MESC-3D, and RepStateDiff outputs are compared with the input image and ground-truth point cloud; PDM is included in the quantitative tables but is not part of this visual panel.

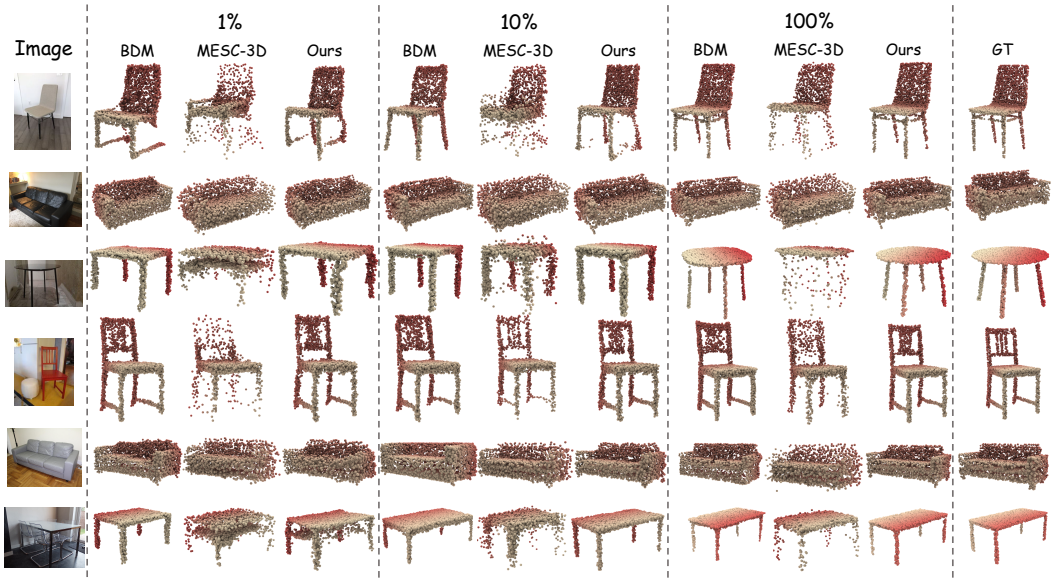

Figure 6: Qualitative Pix3D comparison across 1%, 10%, and 100% training-data regimes. Rows correspond to data regimes, and columns compare table, chair, and sofa reconstructions with input images and ground truth.

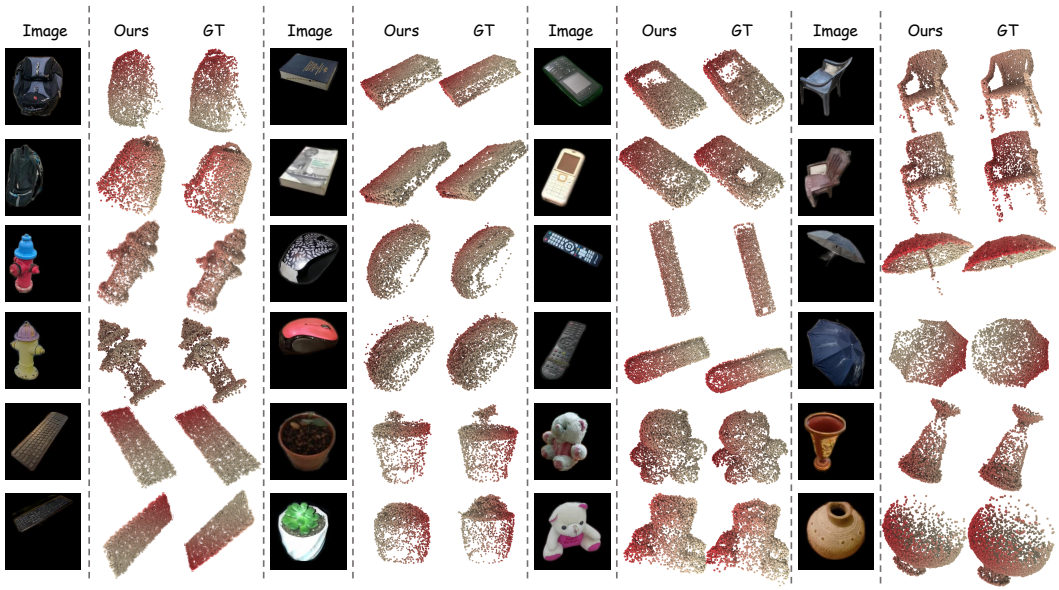

Figure 7: Qualitative CO3Dv2 reconstruction examples from a 12-category subset of the full-data evaluation. Each triplet shows the input image, the RepStateDiff prediction (Ours), and the ground-truth point cloud (GT).