SINGLE-STEP 3D RECONSTRUCTION VIA GEOMETRY-CONDITIONED RECTIFIED TRANSPORT

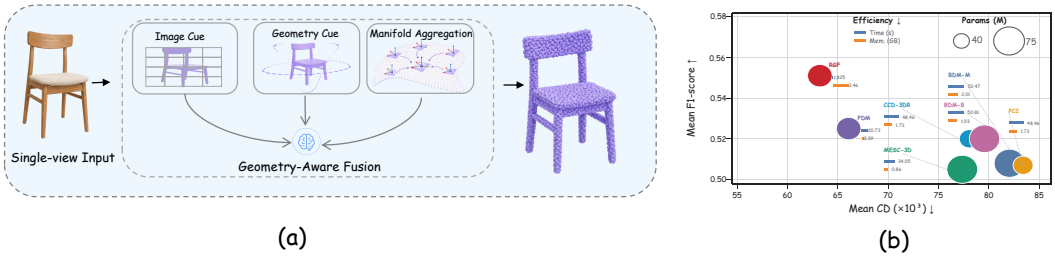
Anonymous authors
Paper under double-blind review

ABSTRACT

The deployment of high-fidelity single-view 3D reconstruction is currently stalled by the limitations of denoising diffusion probabilistic models (DDPMs). Despite their generation quality, DDPMs suffer from prohibitive latency due to iterative sampling and can converge unreliably in data-scarce regimes. We introduce rectified geometric flow (RGF), a method that bypasses these bottlenecks by replacing stochastic diffusion with deterministic, straight-line ODE trajectories. This linearization is not merely an efficiency trick but simplifies the generative mapping, enabling robust learning on scarce data. To prevent structural degradation during rapid generation, we integrate geometric injection as an explicit 3D scaffold and align local features through intrinsic manifold transport. In a matched comparison with the recent Point Diffusion Mamba (PDM) baseline, RGF achieves lower Chamfer Distance (CD) in 9 of 12 category settings and higher F1 in 9 of 12 (with one tie). On an RTX 3090, RGF takes 1.825 s per sample versus 20.73 s for PDM (11.4× faster), while using 3.46 GB rather than PDM's 0.39 GB of GPU memory. Project page: `https://rgf-code.github.io/`.

1 INTRODUCTION

Recovering 3D geometry from a single 2D view is an ill-posed task arising in autonomous robotics Huang et al. (2022); Yao et al. (2023); Kumar et al. (2023) and immersive virtual reality Fan et al. (2022); Chen et al. (2023); Liang et al. (2025). Limited planar observations, self-occlusion, and missing depth cues make the lift to 3D ambiguous. Voxels and meshes Gao et al. (2023); Yang et al. (2023); Zhang et al. (2024); Ha et al. (2025) face resolution or topology limits, whereas point-cloud generators Cui et al. (2024); Zheng et al. (2024); Zhu et al. (2025) provide a flexible alternative. DDPMs Ho et al. (2020) achieve high fidelity by iteratively refining noise to capture high-frequency details beyond deterministic regression. Recent work explores pose-grounded object reconstruction Huang et al. (2026) and single-image scene lifting from sparse visual evidence Wang et al. (2026). Recent diffusion-state-space models, exemplified by Point Diffusion Mamba (PDM), combine diffusion priors with Mamba-based state-space modeling for single-view reconstruction under data scarcity Zhou et al. (2026b).



(a) (b)

Figure 1: (a) Overview of RGF for single-view 3D reconstruction. The framework performs Geometry-Aware Fusion of image cues, geometry cues, and manifold aggregation to guide 3D reconstruction. (b) Quality–efficiency map on ShapeNet-R2N2 with 10% training data. Mean CD, F1-score, parameter count, inference time, and GPU memory are shown for each method, with RGF highlighted in red.

Despite their fidelity, diffusion methods Zhou et al. (2021); Vahdat et al. (2022); Chou et al. (2023); Zheng et al. (2024) face optimization and architectural bottlenecks. Their high-variance stochastic

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.

trajectories require costly iterative sampling Song et al. (2021a); Liu et al. (2023b) and can converge poorly with little data. Existing methods Melas-Kyriazi et al. (2023); Di et al. (2023); Xu et al. (2024) project 2D semantics onto noisy points without an explicit 3D scaffold, leaving depth unresolved; Euclidean aggregation also mixes features from misaligned tangent spaces and weakens topological fidelity.

We propose rectified geometric flow (RGF), which couples geometry-conditioned rectified transport with explicit 3D priors. It transports a state that combines noisy points, image features, a monocular-depth scaffold, and manifold-aligned local features rather than an unconditional Euclidean point set. Geometric injection (GI) back-projects image evidence into 3D, while intrinsic geometry aggregation (IGA) aligns local features through curvature-aware parallel transport. Together, they replace stochastic sampling with deterministic ODE trajectories while preserving single-view constraints. Our contributions are:

- We propose RGF, which learns geometry-conditioned rectified transport (GCRT) on image- and geometry-augmented point states for high-fidelity single-step reconstruction.
- We introduce GI to mitigate depth ambiguity by anchoring image cues to an explicit 3D scaffold.
- We devise IGA, which uses parallel transport on the shape manifold to preserve fine-grained topology beyond Euclidean pooling.
- Against the matched PDM baseline, RGF achieves lower CD in seven and higher F1 in six of nine ShapeNet-R2N2 category/data settings, with an **11.4× inference speedup** (1.825 s vs. 20.73 s); this establishes a favorable latency–quality trade-off, although PDM uses less GPU memory.

2 Related Work

2.1 Single-View Point Cloud Reconstruction

Single-view reconstruction has progressed from deterministic regression Choy et al. (2016); Wang et al. (2018); Wen et al. (2022); Li et al. (2025) to probabilistic generation. Voxels and meshes incur cubic memory costs or impose topological rigidity, whereas point clouds are flexible and efficient. DDPM-based methods Ho et al. (2020); Melas-Kyriazi et al. (2023); Di et al. (2023) improve fidelity by projecting 2D features onto 3D points and stabilizing denoising with recursive refinement Zhou et al. (2025) or Bayesian priors Xu et al. (2024). These projection-based pipelines can leave depth ambiguous, while Euclidean feature aggregation may overlook surface geometry. A directly relevant recent approach, Point Diffusion Mamba (PDM) Zhou et al. (2026b), combines diffusion with local geometric aggregation, Mamba blocks, hierarchical feature integration, and dynamic weighted sampling for data-scarce reconstruction; we use it as a matched recent baseline. RGF instead learns geometry-conditioned rectified transport and distills it to single-step inference, with an explicit depth scaffold and manifold-aware feature alignment. Recent work also grounds reconstruction through joint shape-pose modeling Huang et al. (2026) or sparse image anchors lifted into a 3D scaffold Wang et al. (2026).

2.2 Generative Models

Diffusion models Ho et al. (2020); Song et al. (2021b) are effective for image Rombach et al. (2022), video Blattmann et al. (2023), audio Liu et al. (2023a), and 3D point cloud generation Luo & Hu (2021), but high-fidelity geometric modeling often requires hundreds or thousands of denoising steps. Distillation Song et al. (2023), fast solvers Lu et al. (2022), and subspace constraints Zhou et al. (2021) accelerate sampling, yet do not remove the core trajectory issue: SDE-based reverse paths remain curvilinear and thus require prolonged iterative transport from noise to data. Structure-aware point-cloud generators also incorporate part and symmetry priors to improve global consistency and fine-grained controllability Zhou et al. (2026a).

Flow matching and rectified flow Lipman et al. (2023); Albergo & Vanden-Eijnden (2023); Liu et al. (2023b); Molodyk et al. (2025) regress deterministic ODE velocity fields that induce straight generative trajectories, directly addressing diffusion latency and enabling high-quality image Lipman

et al. (2023) and audio Le et al. (2023) synthesis with fewer steps. Existing 3D variants, however, mainly focus on unconditional generation and treat point clouds as Euclidean particles, overlooking the non-Euclidean manifold structure of surfaces. Their Euclidean transport objective can miss topological details, which motivates our manifold-aware rectified transport framework for single-view reconstruction.

3 METHOD

As illustrated in Fig. 3, the RGF framework bridges the domain gap between 2D images and 3D structures by injecting pixel-aligned features and explicit depth priors directly into the flow state. Building on this injection, we align feature representations across distinct tangent spaces using parallel transport to capture intrinsic local structures. This geometry-aware conditioning drives a geometry-conditioned rectified transport process that deterministically recovers the ground-truth point cloud from noise. The complete training procedure and hyperparameter settings are provided in the supplementary code.

3.1 PRELIMINARY: CONDITIONAL PROBABILITY TRANSPORT

Let $\mathcal{X} \in \mathbb{R}^d$ denote a point cloud with dimension $d = N \times 3$. We consider a time-dependent vector field $v : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ that defines a time-dependent diffeomorphic map, called a flow, $\phi : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$, governed by the ODE:

$$\begin{cases} \dfrac{d}{dt}\phi_t(\mathcal{X}) = v_t(\phi_t(\mathcal{X})), \\ \quad \phi_0(\mathcal{X}) = \mathcal{X}. \end{cases} \tag{1}$$

where $\phi_t(\mathcal{X})$ denotes the state at time t starting from $\mathcal{X}$. This flow transforms a simple prior distribution p_0 to the complex data distribution p_1 and satisfies the push-forward condition.

To efficiently learn the vector field v_t, we adopt the flow matching framework. Instead of simulating the ODE during training, flow matching regresses the neural vector field v_t to a target conditional vector field u_t, which is generated by a specific probability path. Specifically, we utilize conditional flow matching (CFM) with a linear probability path, defined as an interpolation between a source sample $\mathcal{X}_0 \sim p_0$ and a target data sample $\mathcal{X}_1 \sim p_1$:

$$\mathcal{X}_t = (1 - t)\mathcal{X}_0 + t\mathcal{X}_1. \tag{2}$$

This linear path induces a constant conditional vector field $u_t(\mathcal{X}|\mathcal{X}_0, \mathcal{X}_1)$ that drives the probability mass from $\mathcal{X}_0$ to $\mathcal{X}_1$ along a straight line:

$$u_t(\mathcal{X}_t|\mathcal{X}_0, \mathcal{X}_1) = \mathcal{X}_1 - \mathcal{X}_0. \tag{3}$$

Consequently, the neural network v_θ (parameterized by θ) is trained by minimizing the regression objective over the time $t \sim \mathcal{U}[0, 1]$:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,p_0(\mathcal{X}_0),p_1(\mathcal{X}_1)} \left[\|v_\theta(\mathcal{X}_t, t) - (\mathcal{X}_1 - \mathcal{X}_0)\|^2 \right]. \tag{4}$$

where $\mathcal{U}[0, 1]$ denotes the uniform distribution.

3.2 GEOMETRIC INJECTION (GI)

To mitigate the inherent modality disparity, the GI projects pixel-aligned image features onto the 3D flow state and integrates them with explicit geometric priors derived from a depth-based scaffold (see Fig. 2). This explicitly aligns 2D semantics with 3D structures, establishing a unified geometry-aware representation. Specifically, following previous methods Melas-Kyriazi et al. (2023), we first encode the input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ into pixel-aligned feature representations $\mathbf{F}_{img} \in \mathbb{R}^{H\times W\times l}$ using a pre-trained Vision Transformer, where l is the feature dimension. Subsequently, these features are projected onto the flow state $\mathcal{X}_t \in \mathbb{R}^{N\times 3}$ through a rasterization function Π, yielding the baseline semantic conditioning $\mathbf{V} \in \mathbb{R}^{N\times D}$:

$$\mathbf{V} = \{\mathbf{v}_j\}_{j=1}^N = \Pi(\mathbf{F}_{img}, \mathcal{X}_t). \tag{5}$$

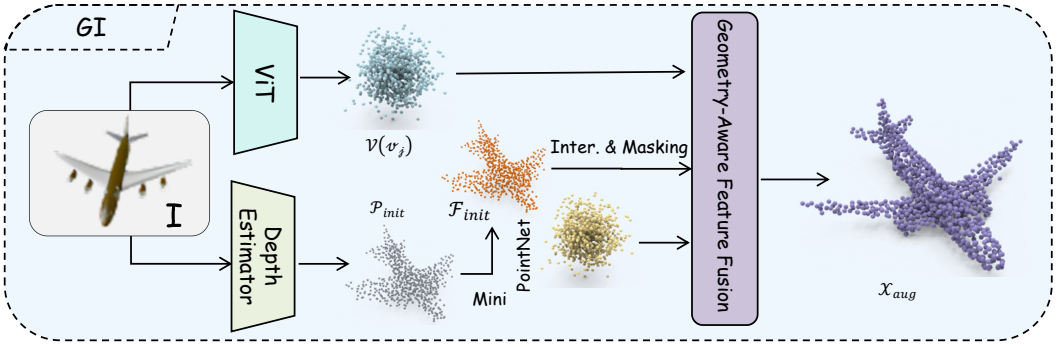


Figure 2: Overview of the Geometric Injection (GI) Module, which fuses pixel-aligned image features with depth-based geometric priors to obtain the geometry-augmented flow state $\mathcal{X}_{aug}$.

From $\mathbf{I}$, we infer a depth map $\mathbf{D} \in \mathbb{R}^{H\times W}$ using a depth estimator Bochkovskii et al. (2025), with depth values d_i. We recover the metric 3D position for each valid homogeneous pixel $\tilde{\mathbf{u}}_i$ via the camera intrinsics $\mathbf{K} \in \mathbb{R}^{3\times3}$, the camera-to-world rotation $\mathbf{R} \in SO(3)$, and translation $\mathbf{t} \in \mathbb{R}^3$, thereby obtaining an explicit geometric prior $\mathcal{P}_{init} \in \mathbb{R}^{M\times3}$:

$$\mathcal{P}_{init} = \{\mathbf{p}_i = \mathbf{R} \cdot \left(d_i \cdot \mathbf{K}^{-1} \cdot \tilde{\mathbf{u}}_i\right) + \mathbf{t}\}_{i=1}^{M}. \tag{6}$$

where M is the number of points in $\mathcal{P}_{init}$. The geometric prior $\mathcal{P}_{init}$ and the dynamic flow state $\mathcal{X}_t$ are both normalized to a unified metric space to ensure consistent interaction.

Subsequently, the geometric prior $\mathcal{P}_{init}$ is encoded into geometric features $\mathbf{F}_{init} \in \mathbb{R}^{M\times C}$ via a mini-PointNet Qi et al. (2017). To transfer these priors to $\mathcal{X}_t$, we employ a density-adaptive k-nearest neighbor interpolation, where the weights $\mathbf{A} = \{\{\alpha_{j,m}\}_{m=1}^{k}\}_{j=1}^{N}$ are computed via the exponential decay of squared Euclidean distances:

$$\alpha_{j,m} = \frac{\exp(-\beta \cdot \|x_t(j) - \mathbf{p}_m\|^2)}{\sum_{\mathbf{p}_m \in \mathcal{N}_k(x_t(j))} \exp(-\beta \cdot \|x_t(j) - \mathbf{p}_m\|^2)}. \tag{7}$$

where $\mathcal{X}_t(j)$ denotes the jth point of $\mathcal{X}_t$, $\mathcal{N}_k(\mathcal{X}_t(j))$ is the set of k nearest neighbors $\mathbf{p}_m$ of $\mathcal{X}_t(j)$ in $\mathcal{P}_{init}$, and m indexes these neighbors from 1 to k. We apply a geometric validity mask based on a distance threshold τ to filter out unreliable sparse regions from priors, and aggregate the valid geometric features as $\mathbf{c}_j \in \mathbb{R}^C$:

$$\begin{cases} \textbf{Mask}_j = \mathbb{I}(\max \|\mathcal{X}_t(j) - \mathbf{p}_m\|^2 < \tau), \\ \mathbf{c}_j = \sum_{m=1}^{k} \textbf{Mask}_j \cdot \alpha_{j,m} \cdot \mathbf{F}_{init}(m). \end{cases} \tag{8}$$

where $\mathbb{I}$ denotes the indicator function: if $\max \|\mathcal{X}_t(j) - \mathbf{p}_m\|^2 < \tau$, we retain the region; otherwise, we discard it. $\mathbf{F}_{init}(m)$ denotes the geometric feature of the mth point in $\mathbf{F}_{init}$. Subsequently, we fuse the semantic conditioning features $\mathbf{v}_j$ and geometric features $\mathbf{c}_j$ via a gated injection mechanism with a learnable weight matrix $\mathbf{W}_g \in \mathbb{R}^{D\times C}$:

$$\mathbf{f}_j = \mathbf{v}_j + \lambda \cdot (\mathbf{W}_g \cdot \mathbf{c}_j). \tag{9}$$

where λ is a learnable scalar and $\mathbf{f}_j \in \mathbb{R}^D$. Finally, the fused features $\mathbf{f}_j$ are concatenated with the spatial coordinates of $\mathcal{X}_t$, yielding the feature-augmented point cloud $\mathcal{X}_{aug} = \{concat(\mathcal{X}_t(j), \mathbf{f}_j)\}_{j=1}^{N} \in \mathbb{R}^{N\times(3+D)}$.

3.3 Intrinsic Geometry Aggregation (IGA)

Given the feature-augmented point cloud $\mathcal{X}_{aug}$ from the previous stage, directly processing all dense points is computationally prohibitive. To reduce complexity while preserving local context, we first use farthest point sampling to select center points $\mathcal{C}$ and K-nearest neighbors to group local patches $\mathcal{G}$, which are then mapped by a lightweight encoder $\xi(\cdot)$ into patch tokens $\mathbf{H} = \{\mathbf{h}_i\}_{i=1}^{S} \in \mathbb{R}^{S\times d_m}$ for the aggregation module:

$$\mathbf{H} = \xi\left(\text{KNN}\left(\text{FPS}(\mathcal{X}_{aug})\right)\right). \tag{10}$$

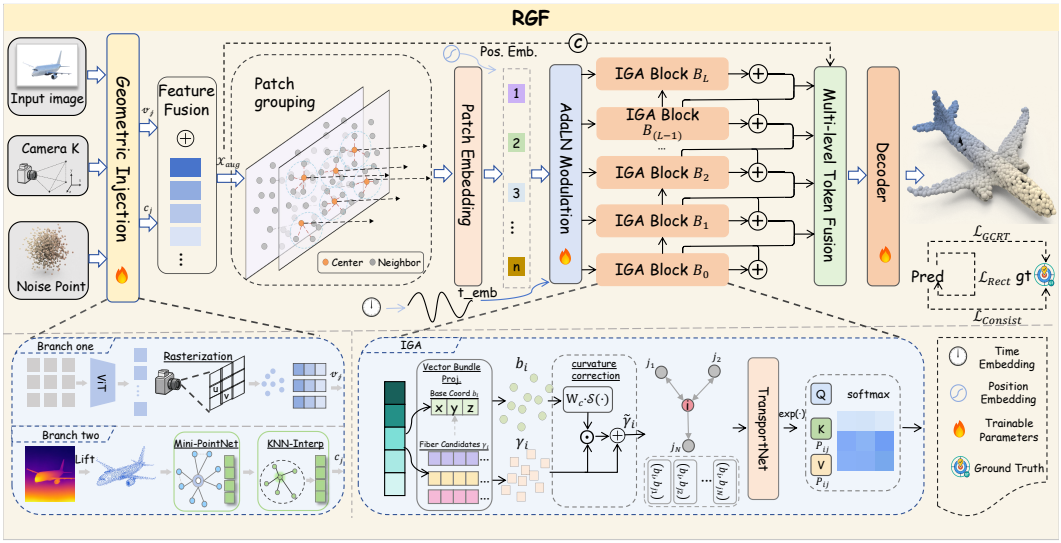


Figure 3: Architecture of RGF and the Intrinsic Geometry Aggregation (IGA) Module. RGF injects image-aligned features and depth-based priors into the point state, then groups the augmented points into local patch tokens. IGA decomposes each token into base and fiber features and aggregates them through curvature-aware correction and parallel transport attention. The aggregated representation is interpolated back to dense points and decoded to predict the rectified transport velocity for efficient single-view 3D reconstruction.

where S is the number of patches in $\mathcal{G}$ and d_m is the feature dimension.

To effectively capture local geometric structures while respecting the manifold topology, we introduce the IGA module. We first define a vector bundle space $\mathcal{E} = \mathcal{B} \times \Gamma$, where each token $\mathbf{h}_i$ is decomposed into a base coordinate $\mathbf{b}_i \in \mathbb{R}^{d_b}$ and a fiber feature $\boldsymbol{\gamma}_i \in \mathbb{R}^{d_f}$ via learnable matrices $\mathbf{W}_b \in \mathbb{R}^{d_b \times d_m}$ and $\mathbf{W}_\gamma \in \mathbb{R}^{d_f \times d_m}$:

$$\mathbf{b}_i = \mathbf{W}_b \cdot \mathbf{h}_i, \quad \boldsymbol{\gamma}_i = \mathbf{W}_\gamma \cdot \mathbf{h}_i. \tag{11}$$

Transporting features across a curved manifold induces holonomy. To compensate for this, we incorporate a curvature correction mechanism. Specifically, we estimate a curvature energy $\mathcal{S}(\mathbf{b}_i) \in \mathbb{R}^{d_b}$ by computing the derivatives of a learned connection field via automatic differentiation. The fiber features are then rectified via a residual update modulated by the curvature transformation weight $\mathbf{W_c} \in \mathbb{R}^{d_f \times d_b}$, yielding $\tilde{\boldsymbol{\gamma}}_i$:

$$\tilde{\boldsymbol{\gamma}}_i = \boldsymbol{\gamma}_i + \tanh(\mathbf{W}_c \cdot \mathcal{S}(\mathbf{b}_i)) \odot \boldsymbol{\gamma}_i. \tag{12}$$

This term $\mathcal{S}(\mathbf{b}_i)$ effectively captures the local geometric distortion intensity around the base coordinates.

To aggregate information, we employ manifold transport attention. We construct a local neighborhood graph based on the Euclidean distance between base coordinates. Recognizing that features reside in distinct tangent spaces, we approximate parallel transport from neighbor j to query i using a connection operator $\mathbf{P}_{ij}$. Crucially, to strictly enforce the structural constraint $\mathbf{P}_{ij} \in SO(d_f)$, we parameterize it as the matrix exponential of a predicted skew-symmetric generator Skew derived solely from the pair of base coordinates:

$$\mathbf{P}_{ij}(\mathbf{b}_i, \mathbf{b}_j) = \exp\left(\mathbf{Skew}(\mathrm{MLP}(\mathbf{b}_i, \mathbf{b}_j))\right). \tag{13}$$

The aggregated feature is then computed by aligning neighbor features to the query's tangent space:

$$\mathbf{h}'_i = \sum_{j \in \mathcal{N}(i)} \mathrm{Softmax}_j\left(\frac{(\mathbf{W}_q \tilde{\boldsymbol{\gamma}}_i)^\top (\mathbf{P}_{ij} \mathbf{W}_k \tilde{\boldsymbol{\gamma}}_j)}{\sqrt{d_f}}\right)(\mathbf{P}_{ij} \mathbf{W}_v \tilde{\boldsymbol{\gamma}}_j). \tag{14}$$

Finally, within each block, we obtain the updated features via a residual update $\mathbf{z}_i = \mathbf{h}_i + \psi(\mathbf{h}'_i)$, where $\psi(\cdot)$ is a feature transformation MLP. To capture both local geometric nuances and global semantic context, we design our backbone by stacking L IGA blocks sequentially. The output of the

l-th block serves as the input to the $(l+1)$-th block, allowing information to propagate across the entire shape manifold. We denote the final output representations from the last block as $\mathbf{Z}$.

To recover the dense point-level velocity field $\mathbf{v}_{pred} \in \mathbb{R}^{N\times 3}$, we utilize these deep aggregated representations $\mathbf{Z}$ to guide the decoding process. For each point $\mathbf{x}_i$, we aggregate latent features from its k-nearest patch centers $\mathcal{N}_k(\mathbf{x}_i)$ using inverse distance weighting. These interpolated features are concatenated with the high-resolution encoding $\mathbf{f}_i$ via a skip connection to preserve details:

$$\mathbf{h}_i^{dense} = \text{Concat}\left(\mathbf{f}_i, \sum_{j\in\mathcal{N}_k(\mathbf{x}_i)} \frac{w_{ij}}{\sum_{m\in\mathcal{N}_k(\mathbf{x}_i)} w_{im}} \mathbf{z}_j\right). \tag{15}$$

A final MLP projects the fused features to the target velocity space:

$$\mathbf{v}_{pred,i} = \text{MLP}(\mathbf{h}_i^{dense}). \tag{16}$$

Table 1: Quantitative comparison on the ShapeNet-R2N2 dataset under different data regimes. Baselines include PC^2 Melas-Kyriazi et al. (2023), CCD-3DR Di et al. (2023), BDM Xu et al. (2024), MESC-3D Li et al. (2025), and PDM Zhou et al. (2026b). Best results are shown in **bold**, and second-best results are underlined.

Method	Chair						Airplane						Car					
	10%		50%		100%		10%		50%		100%		10%		50%		100%	
	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑
PC^2 Melas-Kyriazi et al. (2023)	97.25	0.393	73.58	0.437	65.57	0.464	88.00	0.605	76.39	0.628	65.97	0.655	64.99	0.524	62.59	0.542	64.36	0.547
CCD-3DR Di et al. (2023)	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)	94.94	0.395	71.56	0.446	64.48	0.468	87.75	0.604	73.19	0.629	65.16	0.653	63.53	0.524	60.71	0.549	64.16	0.554
BDM-B Xu et al. (2024)	94.67	0.410	69.99	0.463	64.21	0.485	83.62	0.612	68.66	0.641	59.04	0.660	60.48	0.539	62.58	0.554	65.85	0.559
MESC-3D Li et al. (2025)	101.51	0.381	73.47	0.427	65.69	0.458	74.31	0.611	51.28	0.619	50.54	0.714	56.28	0.522	**44.48**	0.532	51.99	0.602
PDM (ECCV 26) Zhou et al. (2026b)	82.41	0.419	68.85	0.465	62.14	0.488	60.64	0.614	50.14	0.663	48.66	**0.719**	**55.23**	0.542	55.53	0.558	**51.54**	**0.607**
Ours	**76.39**	**0.437**	**58.46**	**0.478**	**56.24**	0.488	**57.30**	**0.669**	**44.71**	**0.711**	**40.60**	0.713	56.00	**0.547**	51.49	**0.581**	53.12	0.578

Table 2: Average performance comparison across all ShapeNet-R2N2 categories under different training data regimes.

Data	BDM Xu et al. (2024)		MESC-3D Li et al. (2025)		**Ours**	
	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑
10%	93.87	0.496	156.66	0.396	**75.98**	**0.517**
50%	69.60	0.547	141.16	0.432	**57.95**	**0.570**
100%	62.81	0.567	135.80	0.454	**54.43**	**0.571**

3.4 Geometry-Conditioned Rectified Transport (GCRT)

To enable rapid, deterministic reconstruction, we depart from the stochastic diffusion paradigm and instantiate rectified transport on geometry-conditioned point states. Instead of applying the vector field in Eq. 4 to raw point coordinates, GCRT augments the current state with the image and geometric evidence required for single-view reconstruction. Given an input image $\mathbf{I}$, the GI module produces pixel-aligned features $\mathbf{F}_I$ and a depth-induced 3D scaffold $\mathcal{P}_I$. The IGA module further provides manifold-aligned local features $\mathbf{\Gamma}_t$ for the current point state. We write the resulting augmented transport state as:

$$\mathcal{X}_{aug}(t) = \mathcal{A}_\psi\left(\mathcal{X}_t, \mathbf{F}_I, \mathcal{P}_I, \mathbf{\Gamma}_t\right). \tag{17}$$

where $\mathcal{A}_\psi$ denotes the geometric augmentation induced by GI and IGA. The transport path follows the linear interpolation from Sec. 3.1, but the velocity is predicted from $\mathcal{X}_{aug}(t)$ instead of $\mathcal{X}_t$ alone. Thus, the learned field is conditioned on the image-derived scaffold throughout the trajectory:

$$\mathcal{L}_{\text{GCRT}}(\theta) = \mathbb{E}_{t,\mathcal{X}_0,\mathcal{X}_1,\mathbf{I}}\left[\left\|v_\theta\left(\mathcal{X}_{aug}(t), t\right) - \left(\mathcal{X}_1 - \mathcal{X}_0\right)\right\|^2\right]. \tag{18}$$

Compared with Eq. 4, the target velocity remains the rectified transport direction, but the domain in which the velocity field is learned is no longer an unconditional Euclidean point space. The flow is evaluated on a state-dependent representation that couples noisy points with projected image semantics, depth-based geometric anchors, and tangent-aligned local features. This makes the transport field reconstruction-aware while preserving the simulation-free training advantage of flow matching.

However, the learned flow may still exhibit curvature due to stochastic couplings or approximation errors, necessitating multiple integration steps to avoid truncation errors. To address this, we employ a geometry-conditioned rectification strategy. We utilize the pre-trained model to simulate ODE trajectories, generating pairs of noise and predicted endpoints $(\mathcal{X}_0, \hat{\mathcal{X}}_1)$. For the rectified interpolation $\hat{\mathcal{X}}_t = (1-t)\mathcal{X}_0 + t\hat{\mathcal{X}}_1$, the geometric augmentation is recomputed on the new state:

$$\hat{\mathcal{X}}_{aug}(t) = \mathcal{A}_\psi\left(\hat{\mathcal{X}}_t, \mathbf{F}_I, \mathcal{P}_I, \hat{\mathbf{\Gamma}}_t\right). \tag{19}$$

We then fine-tune the model to match the constant velocity of the straight line connecting these synthetic pairs under the recomputed geometric state. The objective is defined as:

$$\mathcal{L}_{\text{Rect}}(\theta) = \mathbb{E}_{t,\mathcal{X}_0,\hat{\mathcal{X}}_1,\mathbf{I}}\left[\left\| v_\theta\left(\hat{\mathcal{X}}_{aug}(t), t\right) - \left(\hat{\mathcal{X}}_1 - \mathcal{X}_0\right)\right\|^2\right]. \tag{20}$$

Since the pairing is derived from the model's own flow and the augmentation is recomputed along the rectified trajectory, this procedure straightens the conditional transport under the same image and geometric constraints used for reconstruction. The resulting trajectory therefore reduces discretization error without discarding the depth scaffold or tangent-space alignment.

Finally, to achieve real-time inference, we introduce a trajectory consistency constraint. Since the learned trajectory is nearly straight, the mapping from initial noise to the target shape is deterministic. We enforce self-consistency by optimizing the model to predict the terminal state in a single forward pass. Specifically, we generate a high-quality target $\mathcal{X}_1^{target}$ by integrating the frozen rectified model using a multi-step numerical solver. The current model is then trained to regress this target directly from the initial noise $\mathcal{X}_0$. The objective is to minimize the Chamfer Distance between the one-step prediction $\hat{\mathcal{X}}_1$ and the solver-generated target:

$$\mathcal{L}_{\text{Consist}} = \mathbb{E}_{\mathcal{X}_0}\left[\mathcal{L}_{CD}\left(\hat{\mathcal{X}}_1, \mathcal{X}_1^{target}\right)\right]. \tag{21}$$

where $\hat{\mathcal{X}}_1 = \mathcal{X}_0 + v_\theta(\mathcal{A}_\psi(\mathcal{X}_0, \mathbf{F}_I, \mathcal{P}_I, \mathbf{\Gamma}_0), 0)$ denotes the single-step output. By aligning the one-step prediction with the robust multi-step trajectory under the same geometric augmentation, our framework achieves competitive reconstruction quality with low inference latency.

4 Experiments

4.1 Datasets

We evaluate RGF on ShapeNet-R2N2 Choy et al. (2016) and Pix3D under established protocols Melas-Kyriazi et al. (2023); Xu et al. (2024). ShapeNet-R2N2 includes Airplane, Car, and Chair with provided renderings and cameras, while Pix3D includes Chair, Sofa, and Table under the standard setting. Both use the official 80/20 split; additional low-data and remaining-category results are reported in Appendix B.1.1.

4.2 Implementation Details

RGF is implemented in PyTorch with 224×224 inputs and 4,096 supervised points per object. Pix3D images are cropped by bounding boxes with recalibrated camera matrices. Each category is trained for 100,000 iterations; Euler integration with 1,000 steps is used for flow training and rectification, while final inference is single-step. We report Chamfer Distance (CD) and F-Score@1% on one NVIDIA RTX 3090 GPU. We include Point Diffusion Mamba (PDM) Zhou et al. (2026b) as a recent baseline and evaluate it using the same point count, 224×224 rendering resolution, camera/view protocol, data ratios, dataset splits, metrics, and batch-size-one RTX 3090 measurement setting as the other comparisons.

ShapeNet-R2N2 Table 1 summarizes results across three categories and data regimes, with PDM serving as the most recent matched diffusion-state-space reference. RGF generally leads, especially with scarce supervision. At 10% data, Airplane CD is 57.30 (5.5% below PDM's 60.64) and Chair CD is 76.39 (7.3% below PDM's 82.41). Even on Car, RGF reaches 56.00 CD at 10% data, although PDM is slightly better at 55.23. Across the nine ShapeNet settings, RGF is better than PDM in seven

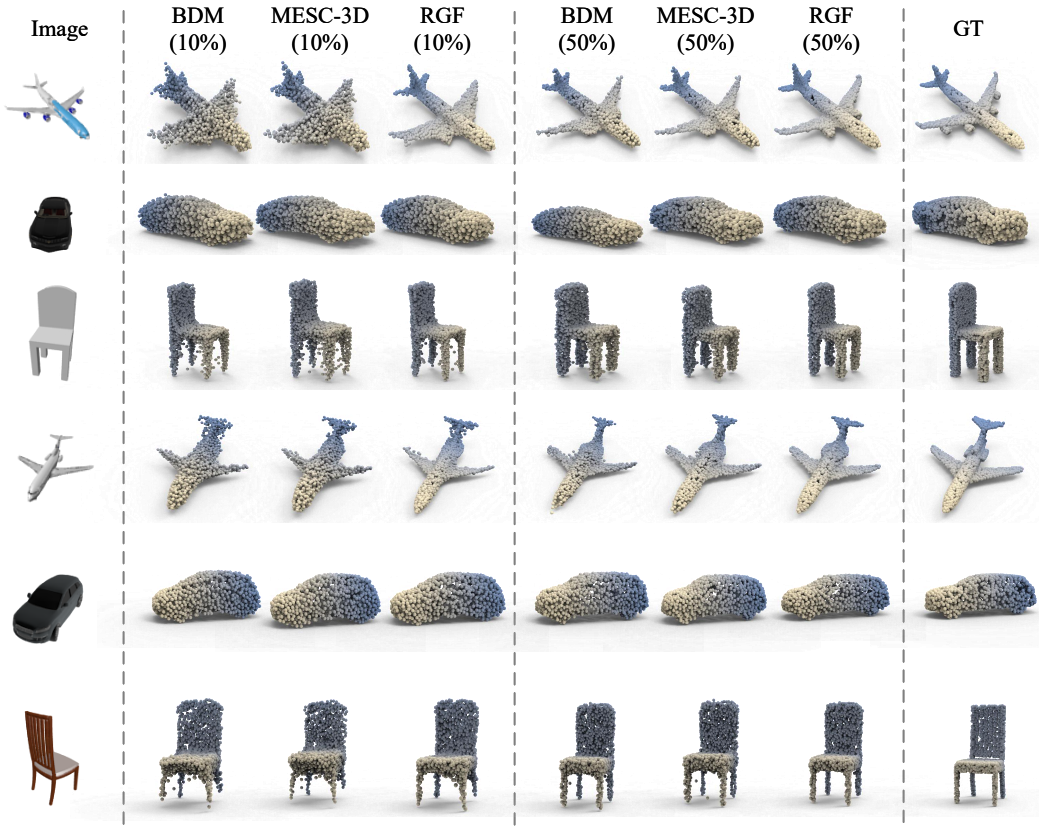


Figure 4: Qualitative ShapeNet-R2N2 comparison under 10% (Cols 2-4) and 50% (Cols 5-7) data regimes relative to Ground Truth (Col 8).

CD cells and six F1 cells, with a tie on Chair at 100% data; PDM is stronger on Car CD at 10%/100% and on Airplane/Car F1 at 100%. Table 2 reports category averages over all ShapeNet-R2N2 classes, while the earlier baselines in the table provide historical context. These results suggest that geometry-conditioned transport and explicit scaffolding stabilize low-data learning and reduce overfitting in iterative models.

Pix3D Table 3 evaluates real-world Pix3D with varied appearance, imperfect alignment, and category-specific structures. RGF reaches 63.69 CD on Chair, 19.7% below PDM's 79.28, while recovering thin furniture structures. For Table, our method lowers CD to 140.18 (23.9% below PDM's 184.35). PDM obtains the best Sofa CD (41.43), while RGF retains the highest F1 in all categories and the best CD on Chair and Table. These gains indicate that image cues, depth geometry, and manifold aggregation transfer from synthetic training to real objects, although the Sofa CD result shows that the quality advantage is not uniform across every metric.

4.3 Qualitative and Efficiency Analysis

Fig. 4 shows that RGF preserves global structure and thin details on ShapeNet-R2N2. At both 10% and 50% supervision, silhouettes and part layouts match the targets, while chair legs and airplane wings are less fragmented than in competing outputs. This is consistent with Table 2: the explicit scaffold preserves recognizable structure under limited data rather than relying only on image-conditioned semantics. Additional qualitative results, including Pix3D in Fig. 7, are provided in Appendix B.2 and Appendix B.1.1. On Pix3D, geometry-aware conditioning transfers to real images with imperfect alignment and appearance variation, preserving object extent and local surfaces across categories. The PDM comparison provides a recent diffusion-state-space reference: PDM is numerically competitive and attains the best Sofa CD, whereas RGF preserves better point coverage, as reflected by the higher F1 scores across all Pix3D categories.

Table 3: Pix3D performance comparison, including the recent PDM baseline Zhou et al. (2026b), using full training data.

Method	Chair		Sofa		Table	
	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑
PC^2 (CVPR 23) Melas-Kyriazi et al. (2023)	115.94	0.443	47.17	0.445	202.77	0.397
CCD-3DR (ArXiv 23) Di et al. (2023)	111.42	0.456	44.91	0.450	196.28	0.418
BDM-M (CVPR 24) Xu et al. (2024)	113.40	0.449	44.50	0.451	202.08	0.413
BDM-B (CVPR 24) Xu et al. (2024)	110.60	0.455	45.05	0.455	186.46	<u>0.429</u>
MESC-3D (CVPR 25) Li et al. (2025)	91.36	0.370	<u>41.98</u>	0.284	206.16	0.306
PDM (ECCV 26) Zhou et al. (2026b)	<u>79.28</u>	<u>0.499</u>	**41.43**	<u>0.463</u>	<u>184.35</u>	0.422
Ours	**63.69**	**0.519**	42.96	**0.474**	**140.18**	**0.436**

Table 4: Efficiency and ablation analyses. (a) Model cost. (b) Component ablation.

Method	Params	Time	Mem.
PC^2/CCD-3DR Melas-Kyriazi et al. (2023); Di et al. (2023)	**47.41**	48.46	1.73
BDM-M Xu et al. (2024)	74.82	52.47	2.01
BDM-B Xu et al. (2024)	73.78	50.81	1.93
MESC-3D Li et al. (2025)	74.84	34.05	<u>0.86</u>
PDM (ECCV 26) Zhou et al. (2026b)	<u>56.34</u>	<u>20.73</u>	**0.39**
Ours	57.78	**1.825**	3.46

(a) Parameters (M), runtime (s), and memory (GB).

Setting	Components			Metrics	
	GCRT	GI	IGA	CD ↓	F1 ↑
Full Model	✓	✓	✓	**44.71**	**0.711**
w/o GCRT	✗	✓	✓	72.23	0.618
w/o IGA	✓	✓	✗	47.75	0.676
w/o GI	✓	✗	✓	48.59	0.632
w/ GI	✗	✓	✗	46.77	0.697
w/ IGA	✗	✗	✓	53.12	0.635
w/ GCRT	✓	✗	✗	49.89	0.642

(b) Component ablation.

Table 4(a) reports efficiency on one NVIDIA RTX 3090 GPU. RGF uses 57.78M parameters, whereas PDM Zhou et al. (2026b) uses 56.34M, showing that the speed advantage does not come from a substantially smaller network. GCRT enables deterministic single-step inference in **1.825s**, compared with 20.73s for PDM, 34.05s for MESC-3D, and 48.46–52.47s for the other baselines; RGF is approximately $11.4\times$ faster than PDM, in addition to the $18.7\times$ and $27.8\times$ speedups over MESC-3D and BDM under the earlier reported measurements. PDM uses only 0.39 GB of GPU memory, whereas RGF uses 3.46 GB because GI and IGA retain geometric features and transport operators during the forward pass. Thus, RGF deliberately trades a higher memory footprint for substantially lower latency; this trade-off favors latency-sensitive reconstruction but should be considered when deploying on memory-constrained devices. The matched PDM comparison is the primary recent reference, while the other rows provide historical context.

4.4 Ablation Studies

Table 4(b) validates GCRT, GI, and IGA by removing or isolating each component. Removing GCRT raises CD from 44.71 to 72.23, confirming the role of rectification and trajectory consistency; without them, curved stochastic transport is difficult to solve accurately in only a few steps. Removing IGA raises CD to 47.75, showing that tangent-space alignment preserves local geometry, whereas Euclidean aggregation can smooth fine structures. The GI-only variant outperforms w/o GI because the scaffold $\mathcal{P}_{init}$ supplies 3D guidance when image semantics are ambiguous; without GI, only 2D features remain. Noisy depth priors explain why the full model performs best: GI anchors the shape, IGA corrects local inconsistencies, and GCRT refines transport.

5 Conclusion

We presented RGF, a geometry-conditioned rectified transport framework for single-view 3D reconstruction. The key insight is to rectify transport in a state that carries explicit 3D evidence: GCRT enables deterministic single-step prediction, GI provides a depth-derived scaffold, and IGA performs curvature-aware local alignment. In matched comparisons with PDM, RGF lowers CD in 9 of 12 category settings and raises F1 in 9 of 12 (with one tie), while reducing RTX 3090 inference time from 20.73s to 1.825s; PDM uses less memory and leads on several individual metrics. The ablation study confirms that all three components are complementary. Evaluation remains limited to object-centric benchmarks and depends on monocular depth and camera calibration; extending RGF to cluttered, unconstrained scenes remains future work.

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

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2209.15571.

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.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22563–22575, 2023. doi:10.1109/CVPR52729.2023.02161.

Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *ICLR*, 2025. URL https://arxiv.org/abs/2410.02073.

Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21674–21683, 2023. doi:10.1109/CVPR52729.2023.02076.

Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2262–2272, 2023. doi:10.1109/ICCV51070.2023.00215.

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 *ECCV*, 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, et al. Lam3d: Large image-point clouds alignment model for 3d reconstruction from single image. *NIPS*, 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. *arXiv preprint arXiv:2308.07837*, 2023. URL https://arxiv.org/abs/2308.07837.

Lue Fan, Feng Wang, Naiyan Wang, and Zhao-Xiang Zhang. Fully sparse 3d object detection. *Advances in Neural Information Processing Systems*, pp. 351–363, 2022. URL https://arxiv.org/abs/2207.10035.

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

Junna Gao, Dehui Kong, Shaofan Wang, Jinghua Li, and Baocai Yin. Cignet: Category-and-intrinsic-geometry guided network for 3d coarse-to-fine reconstruction. *Neurocomputing*, 2023. doi:10.1016/j.neucom.2023.126607.

Juhyung Ha, Vibhas Kumar Vats, Soon-heung Jung, Alimoor Reza, and David J Crandall. Hvpunet: Hybrid-voxel point-cloud upsampling network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 29153–29162, 2025. doi:10.1109/ICCV51701.2025.02707.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NIPS*, 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*, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Huang_CUPID_Generative_3D_Reconstruction_via_Joint_Object_and_Pose_Modeling_CVPR_2026_paper.pdf.

Keli Huang, Botian Shi, Xiang Li, Xin Li, Siyuan Huang, and Yikang Li. Multi-modal sensor fusion for auto driving perception: A survey. *arXiv preprint arXiv:2202.02703*, 2022. URL https://arxiv.org/abs/2202.02703.

Chenru Jiang, Chengrui Zhang, Xi Yang, Jie Sun, and Kaizhu Huang. Consistency diffusion models for single-image 3d reconstruction with priors. *PAMI*, 2025. URL https://arxiv.org/abs/2501.16737.

Pengfei Jin, Tianhao Lai, Rongjie Lai, and Bin Dong. Nptc-net: Narrow-band parallel transport convolutional neural networks on point clouds. *Journal of Scientific Computing*, 2022. doi:10.1007/s10915-021-01699-2.

Varun Ravi Kumar, Ciarán Eising, Christian Witt, and Senthil Kumar Yogamani. Surround-view fisheye camera perception for automated driving: Overview, survey & challenges. *IEEE Transactions on Intelligent Transportation Systems*, pp. 3638–3659, 2023. doi:10.1109/TITS.2023.3235057.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, pp. 14005–14034, 2023. URL https://arxiv.org/abs/2306.15687.

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 Computer Vision and Pattern Recognition Conference*, pp. 16912–16921, 2025. doi:10.1109/CVPR52734.2025.01576.

Ao Liang, Lingdong Kong, Dongyue Lu, Youquan Liu, Jian Fang, Huaici Zhao, and Wei Tsang Ooi. Perspective-invariant 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27725–27738, 2025. doi:10.1109/ICCV51701.2025.02574.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.02747.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. AudioLDM: Text-to-audio generation with latent diffusion models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023a. URL https://arxiv.org/abs/2301.12503.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023b. URL https://arxiv.org/abs/2209.03003.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, pp. 5775–5787, 2022. URL https://arxiv.org/abs/2206.00927.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, 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.

Petr Molodyk, Jaemoo Choi, David W Romero, Ming-Yu Liu, and Yongxin Chen. Mfm-point: Multi-scale flow matching for point cloud generation. *arXiv preprint arXiv:2511.20041*, 2025. URL https://arxiv.org/abs/2511.20041.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. doi:10.1109/CVPR.2017.16.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022. doi:10.1109/CVPR52688.2022.01042.

Stefan C. Schonsheck, Bin Dong, and Rongjie Lai. Parallel transport convolution: Deformable convolutional networks on manifold-structured data. *SIAM Journal on Imaging Sciences*, 15(1): 367–386, 2022. doi:10.1137/21M1407616.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 32211–32252, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *CVPR*, 2024. doi:10.1109/CVPR52733.2024.00972.

Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *NIPS*, 2022. doi:10.52202/068431-0728.

Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 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 *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.19766.

Xin Wen, Junsheng Zhou, Yu-Shen Liu, Hua Su, Zhen Dong, and Zhizhong Han. 3d shape reconstruction from 2d images with disentangled attribute flow. In *CVPR*, 2022. doi:10.1109/CVPR52688.2022.00378.

Lemeng Wu, Dilin Wang, Chengyue Gong, Xingchao Liu, Yunyang Xiong, Rakesh Ranjan, Raghuraman Krishnamoorthi, Vikas Chandra, and Qiang Liu. Fast point cloud generation with straight flows. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9445–9454, 2023. doi:10.1109/CVPR52729.2023.00911.

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.

Xianghui Yang, Guosheng Lin, and Luping Zhou. Single-view 3d mesh reconstruction for seen and unseen categories. *TIP*, 2023. doi:10.1109/TIP.2023.3279661.

Shanliang Yao, Runwei Guan, Xiaoyu Huang, Zhuoxiao Li, Xiangyu Sha, Yong Yue, Eng Gee Lim, Hyungjoon Seo, Ka Lok Man, Xiaohui Zhu, et al. Radar-camera fusion for object detection and semantic segmentation in autonomous driving: A comprehensive review. *IEEE Transactions on Intelligent Vehicles*, pp. 2094–2128, 2023. doi:10.1109/TIV.2023.3307157.

Shijie Zhang, Boyan Jiang, Keke He, Junwei Zhu, Ying Tai, Chengjie Wang, Yinda Zhang, and Yanwei Fu. T-pixel2mesh: Combining global and local transformer for 3d mesh generation from a single image. In *ICASSP*, 2024. doi:10.1109/ICASSP48485.2024.10446395.

Xiao Zheng, Xiaoshui Huang, Guofeng Mei, Yuenan Hou, Zhaoyang Lyu, Bo Dai, Wanli Ouyang, and Yongshun Gong. Point cloud pre-training with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22935–22945, 2024. doi:10.1109/CVPR52733.2024.02164.

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 *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 *ICCV*, 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. *TIP*, 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 *CVPR*, pp. 11789–11798, 2025. doi:10.1109/CVPR52734.2025.01101.

Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *CVPR*, 2024. doi:10.1109/CVPR52733.2024.00983.

SUPPLEMENTARY MATERIAL

Technical_Appendix.pdf The "Technical Appendix.pdf" of RGF includes:

- Section A Highlights.
- Section B Supplementary experimental analysis.
  - Sec. B.1 ShapeNet-R2N2
  - Sec. B.1.1 Additional ShapeNet-R2N2 Results
  - Sec. B.2 Pix3D
  - Sec. B.3 Model Efficiency
  - Sec. B.4 Train Curves
- Section C Additional Validation and Clarifications.
  - Sec. C.1 Evaluation protocol and controlled-analysis scope
  - Sec. C.2 Input-prior provenance and implementation conventions
  - Sec. C.3 Efficiency accounting and speedup definitions
  - Sec. C.4 Single-step quality, stability, and baselines
  - Sec. C.5 Generalization and recent comparisons
  - Sec. C.6 Compute, depth robustness, and data efficiency
  - Sec. C.7 Controlled-analysis boundaries and open diagnostics
  - Sec. C.8 Cross-representation comparison scope
  - Sec. C.9 Scope, limitations, and deployment considerations
- Section D MESC-3D Reproduction Notes.
  - Sec. D.1 Practical Reproduction Issues
  - Sec. D.2 Reproduction Setup
- Section E Limitations and Broader Impact.
- Section F Comparison with Related Methods.
  - Sec. F.1 Rectified Flow
  - Sec. F.2 PSF & MFM-Point
  - Sec. F.3 Transport-aware point processing
  - Sec. F.4 PC^2, CCD-3DR & BDM
  - Sec. F.5 Consistency-oriented and feed-forward reconstruction
  - Sec. F.6 MESC-3D
  - Sec. F.7 GECO
- Section G Preliminary.
  - Sec. G.1 Conditional Point Cloud Diffusion Model
  - Sec. G.2 Quantitative Results
- Section H Theoretical Foundations of IGA.
  - Sec. H.1 Bundle notation and discrete operator
  - Sec. H.2 Transport and attention guarantees
  - Sec. H.3 Curvature correction and stability
  - Sec. H.4 Scope of the guarantees
- Section I Visualization of Geometric Priors.

A HIGHLIGHTS

The highlights of our paper are summarized in the following:

- **Problem motivation and shift to deterministic flow:** Identifies the latency and convergence bottlenecks of stochastic diffusion models (DDPMs) in data-scarce regimes, proposing Geometry-Conditioned Rectified Transport (GCRT) to replace iterative sampling with deterministic, straight-line ODE trajectories.
- **Explicit geometric guidance:** Introduces a Geometric Injection (GI) mechanism that leverages monocular depth estimates to back-project 2D semantics into a 3D scaffold, explicitly resolving the depth ambiguity and structural degradation inherent in standard direct projection methods.
- **Manifold-aware feature aggregation:** Proposes Intrinsic Geometry Aggregation (IGA) based on vector bundle theory, which utilizes parallel transport to align features across distinct tangent spaces, thereby mitigating geometric dissonance and preserving fine-grained topology better than Euclidean attention.
- **Data efficiency and reconstruction quality:** Demonstrates strong reconstruction quality on ShapeNet-R2N2 and Pix3D under limited supervision. In the matched PDM comparison, RGF lowers CD in seven of nine ShapeNet-R2N2 category/data settings and two of three Pix3D categories; it also attains higher F1 in six ShapeNet settings and all three Pix3D categories, while PDM remains better on selected Car/Sofa CD and full-data F1 values.
- **Single-step, low-latency inference:** On the matched RTX 3090 protocol, RGF takes 1.825s/sample versus 20.73s for PDM, an 11.4$\times$ speedup, while using more GPU memory (3.46GB vs. 0.39GB). This speed advantage comes from deterministic rectified transport and single-step prediction rather than a substantially smaller parameter count.

Table 5: Performance improvement of RGF over ShapeNet-R2N2 comparators, including PDM. "-" indicates that RGF is inferior on that metric.

Method	Chair						Airplane						Car					
	10%		50%		100%		10%		50%		100%		10%		50%		100%	
	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑
PC^2	97.25	0.393	73.58	0.437	65.57	0.464	88.00	0.605	76.39	0.628	65.97	0.655	64.99	0.524	62.59	0.542	64.36	0.547
	21.45%	11.20%	20.55%	9.38%	14.23%	5.17%	34.89%	10.58%	41.47%	13.22%	38.46%	8.85%	13.83%	4.39%	17.73%	7.20%	17.46%	5.67%
CCD-3DR	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
	14.92%	4.55%	7.40%	0.84%	3.81%	-	29.51%	9.31%	38.29%	11.97%	35.32%	9.52%	11.29%	3.01%	17.29%	5.64%	14.16%	2.85%
BDM-M	94.94	0.395	71.56	0.446	64.48	0.468	87.75	0.604	73.19	0.629	65.16	0.653	63.53	0.524	60.71	0.549	64.16	0.554
	19.54%	10.63%	18.31%	7.17%	12.78%	4.27%	34.70%	10.76%	38.91%	13.04%	37.69%	9.19%	11.85%	4.39%	15.19%	5.83%	17.21%	4.33%
BDM-B	94.67	0.410	69.99	0.463	64.21	0.485	83.62	0.612	68.66	0.641	59.04	0.660	60.48	0.539	62.58	0.554	65.85	0.559
	19.31%	6.59%	16.47%	3.24%	12.41%	0.62%	31.48%	9.31%	34.88%	10.92%	31.23%	8.03%	7.41%	1.48%	17.72%	4.87%	19.33%	3.40%
MESC-3D	101.51	0.381	73.47	0.427	65.69	0.458	74.31	0.611	51.28	0.619	50.54	0.714	56.28	0.522	**44.48**	0.532	51.99	0.602
	24.75%	14.70%	20.43%	11.94%	14.39%	6.55%	22.89%	9.49%	12.81%	14.86%	19.67%	-	0.50%	4.79%	-	9.21%	-	-
PDM (ECCV 26) Zhou et al. (2026b)	82.41	0.419	68.85	0.465	62.14	0.488	60.64	0.614	50.14	0.663	48.66	**0.719**	**55.23**	0.542	55.53	0.558	**51.54**	**0.607**
	7.30%	4.30%	15.09%	2.80%	9.49%	0.00%	5.51%	8.96%	10.83%	7.24%	16.56%	-	-	0.92%	7.28%	4.12%	-	-
RGF	**76.39**	**0.437**	**58.46**	**0.478**	**56.24**	0.488	**57.30**	**0.669**	**44.71**	**0.711**	**40.60**	0.713	56.00	**0.547**	51.49	**0.581**	53.12	0.578

B Supplementary of Experiment Analysis

B.1 ShapeNet-R2N2 Dataset: Single-View 3D Reconstruction and Low-Data Robustness

We evaluate RGF against PC^2 Melas-Kyriazi et al. (2023), CCD-3DR Di et al. (2023), BDM-M/BDM-B Xu et al. (2024), MESC-3D Li et al. (2025), and the matched PDM reference on ShapeNet-R2N2 across data scales and categories. As shown in Table 5, RGF remains strong in low-data regimes. Against PDM, RGF lowers CD in seven of nine ShapeNet-R2N2 category/data settings and raises F1 in six; PDM leads on selected Car CD and full-data F1 results, so the metrics are considered separately. The earlier rows provide complementary historical comparisons rather than the primary current benchmark.

Comparison with PC^2 (CVPR 2023): PC^2 relies on projection-conditioned diffusion, which struggles with "ray ambiguity" in sparse views. In contrast, RGF integrates the **Geometry-Conditioned Rectified Transport (GCRT)** with **Geometric Injection (GI)**. The deterministic ODE trajectories of GCRT simplify the learning process, while GI anchors 2D semantics to explicit 3D scaffolds. This design leads to RGF's 21.45% CD reduction and 11.20% F1 improvement in the low-data Chair scenario compared to PC^2.

Comparison with CCD-3DR (arXiv 2023): CCD-3DR employs contrastive distillation but lacks mechanisms to strictly enforce geometric consistency on manifolds. RGF's **Intrinsic Geometry Aggregation (IGA)** utilizes parallel transport to align features within the tangent space, preserving geometric fidelity better than Euclidean attention. Consequently, RGF achieves a 29.51% CD reduction in the low-data Airplane category relative to CCD-3DR.

Comparison with BDM-M/BDM-B (CVPR 2024): BDM utilizes stochastic SDE-based diffusion, which requires extensive data to converge and is prone to structural degradation in data-scarce regimes. RGF's linearized flow ensures robust convergence even with limited samples. This advantage is evident in the Chair category (10% data), where RGF outperforms BDM-M by reducing CD by 19.54% and increasing F1 by 10.63%.

Comparison with MESC-3D (CVPR 2025): While MESC-3D focuses on semantic mining, it can overlook the underlying manifold topology. RGF combines explicit depth priors (GI) with manifold-aware aggregation (IGA), enabling it to capture fine-grained structures that semantic-based methods may miss. As an additional pairwise comparison, this gives a 24.75% Chair CD reduction and a 14.70% F1 improvement over our MESC-3D reproduction at 10% data. The matched comparison with the more recent PDM baseline is summarized separately in Table 14 and is the primary reference for the paper's main claims.

B.1.1 Additional ShapeNet-R2N2 Results

Table 6 reports two supplementary ShapeNet-R2N2 analyses: GI threshold sensitivity on Airplane with 50% training data, and additional 1%/25% low-data results on Airplane, Car, and Chair. The threshold sweep is stable from 0.001 to 0.01, suggesting that the lifted depth prior acts as a soft geometric scaffold. Table 7 gives the remaining ten-category results for BDM Xu et al. (2024), MESC-3D Li et al. (2025), and RGF. Best results are in **bold**; second-best results are underlined.

Table 6: Additional ShapeNet-R2N2 analyses. (a) GI threshold ablation on ShapeNet-R2N2 Airplane with 50% training data. (b) Additional low-data ShapeNet-R2N2 results on Airplane, Car, and Chair with 1% and 25% training data.

(a) GI threshold ablation.

GI Threshold	CD ↓	F1 ↑
0.001	**44.29**	**0.713**
0.0025	44.71	0.711
0.005	44.81	0.710
0.01	44.74	0.709

(b) Low-data results.

Method	Data	Airplane		Car		Chair	
		CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑
BDM	1%	115.99	**0.588**	65.27	**0.511**	**107.64**	**0.391**
	25%	69.98	0.635	59.91	0.544	79.78	0.435
MESC-3D	1%	140.54	0.501	137.73	0.435	235.28	0.244
	25%	123.82	0.562	130.32	0.483	199.81	0.268
Ours	1%	**84.32**	0.580	**62.02**	0.495	120.34	0.351
	25%	**50.63**	**0.693**	**53.56**	**0.564**	**64.51**	**0.464**

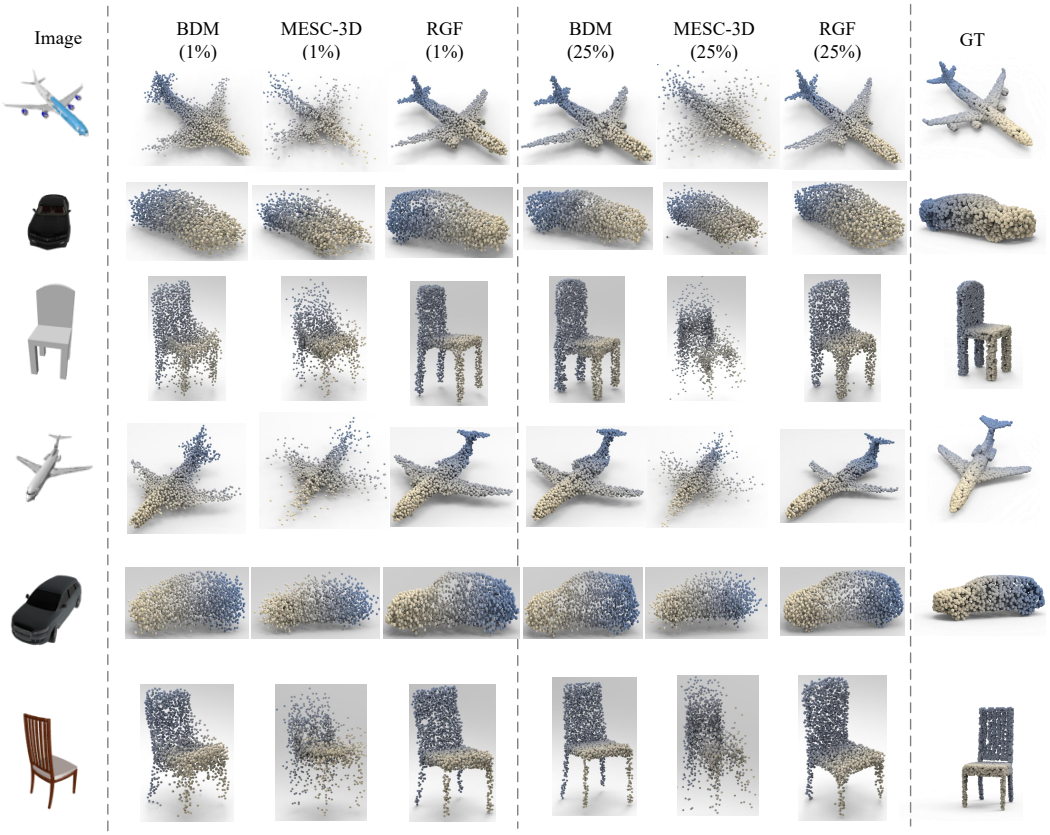


Figure 5: Qualitative comparison of BDM, MESC-3D, and RGF on ShapeNet-R2N2 Airplane, Car, and Chair under 1% and 25% training data.

Table 8 summarizes the dataset split statistics and category-averaged reconstruction performance on ShapeNet-R2N2 and Pix3D. The average metrics for both ShapeNet-R2N2 and Pix3D are computed over the categories on which the corresponding experiments have been conducted.

Table 7: Complete ShapeNet-R2N2 results on the remaining ten categories. (a) Bench, Cabinet, Display, Lamp, and Loudspeaker. (b) Rifle, Sofa, Table, Telephone, and Watercraft.

(a) Bench, Cabinet, Display, Lamp, and Loudspeaker.

Method	Data	Bench		Cabinet		Display		Lamp		Loudspeaker	
		CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑
BDM	10%	80.93	0.542	97.98	0.341	119.32	**0.519**	211.54	0.341	118.87	0.333
	50%	54.60	0.630	57.43	0.442	89.98	0.538	157.80	0.429	89.64	0.395
	100%	**43.21**	**0.677**	55.24	0.459	83.03	**0.554**	131.44	0.453	84.45	0.406
MESC-3D	10%	180.38	0.337	203.14	0.250	175.57	0.381	212.86	0.232	240.13	0.247
	50%	160.14	0.427	185.10	0.282	168.78	0.399	190.40	0.301	226.67	0.252
	100%	151.98	0.421	175.53	0.297	161.69	0.422	181.07	0.302	219.77	0.249
Ours	10%	**67.01**	**0.554**	**71.40**	**0.390**	**86.74**	0.497	**166.25**	**0.390**	**105.69**	**0.335**
	50%	**48.02**	**0.664**	**47.31**	**0.473**	**73.70**	**0.553**	**122.87**	**0.481**	**77.14**	**0.404**
	100%	44.04	0.655	**43.19**	**0.506**	**63.06**	0.549	**103.97**	**0.506**	**68.72**	**0.430**

(b) Rifle, Sofa, Table, Telephone, and Watercraft.

Method	Data	Rifle		Sofa		Table		Telephone		Watercraft	
		CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑
BDM	10%	**39.76**	**0.756**	65.39	0.438	126.40	0.433	62.98	**0.633**	58.35	0.554
	50%	**29.39**	**0.770**	46.97	0.487	78.69	0.522	49.38	0.661	49.65	0.585
	100%	**26.12**	**0.785**	44.48	0.497	71.80	**0.545**	44.13	**0.681**	43.59	0.606
MESC-3D	10%	92.80	0.727	166.62	0.329	239.83	0.217	144.11	0.490	149.00	0.423
	50%	88.88	0.741	160.44	0.353	212.38	0.278	132.08	0.535	140.93	0.471
	100%	88.43	0.753	158.37	0.373	192.91	0.306	130.59	0.531	136.83	0.475
Ours	10%	43.43	**0.756**	**63.59**	**0.450**	**79.20**	**0.507**	**60.30**	0.630	**54.41**	**0.562**
	50%	30.43	0.762	**45.22**	**0.502**	**68.94**	**0.530**	**42.98**	**0.666**	**42.11**	**0.608**
	100%	41.13	0.669	**41.60**	**0.518**	**69.20**	0.521	**43.31**	0.679	**39.45**	**0.607**

Table 8: Dataset split statistics and category-averaged reconstruction performance on ShapeNet-R2N2 and Pix3D.

Dataset Split Statistics				Category-Averaged Performance			
Category	Train	Test	Total	Method	Data	CD ↓	F1 ↑
ShapeNet-R2N2				**ShapeNet-R2N2 Average**			
Airplane	3236	809	4045		10%	93.87	0.496
Bench	1452	363	1815	BDM	50%	69.60	0.547
Cabinet	1257	315	1572		100%	62.81	0.567
Car	5997	1500	7497		10%	156.66	0.396
Chair	5420	1355	6775	MESC-3D	50%	141.16	0.432
Display	876	219	1095		100%	135.80	0.454
Lamp	1854	464	2318		10%	**75.98**	**0.517**
Loudspeaker	1292	323	1615	**Ours**	50%	**57.95**	**0.570**
Rifle	1898	475	2373		100%	**54.43**	**0.571**
Sofa	2538	635	3173				
Table	6800	1700	8500				
Telephone	841	211	1052				
Watercraft	1550	388	1938				
Pix3D				**Pix3D Average**			
Chair	3071	768	3839	BDM	Full	114.04	0.446
Sofa	1557	390	1947	MESC-3D	Full	113.17	0.320
Table	1496	374	1870	**Ours**	Full	**82.28**	**0.476**

B.2 PIX3D DATASET: GENERALIZATION IN REAL-SCENE SINGLE-VIEW 3D RECONSTRUCTION

We assess RGF against PC^2 Melas-Kyriazi et al. (2023), CCD-3DR Di et al. (2023), BDM-M/BDM-B Xu et al. (2024), MESC-3D Li et al. (2025), and PDM on the challenging real-world Pix3D dataset

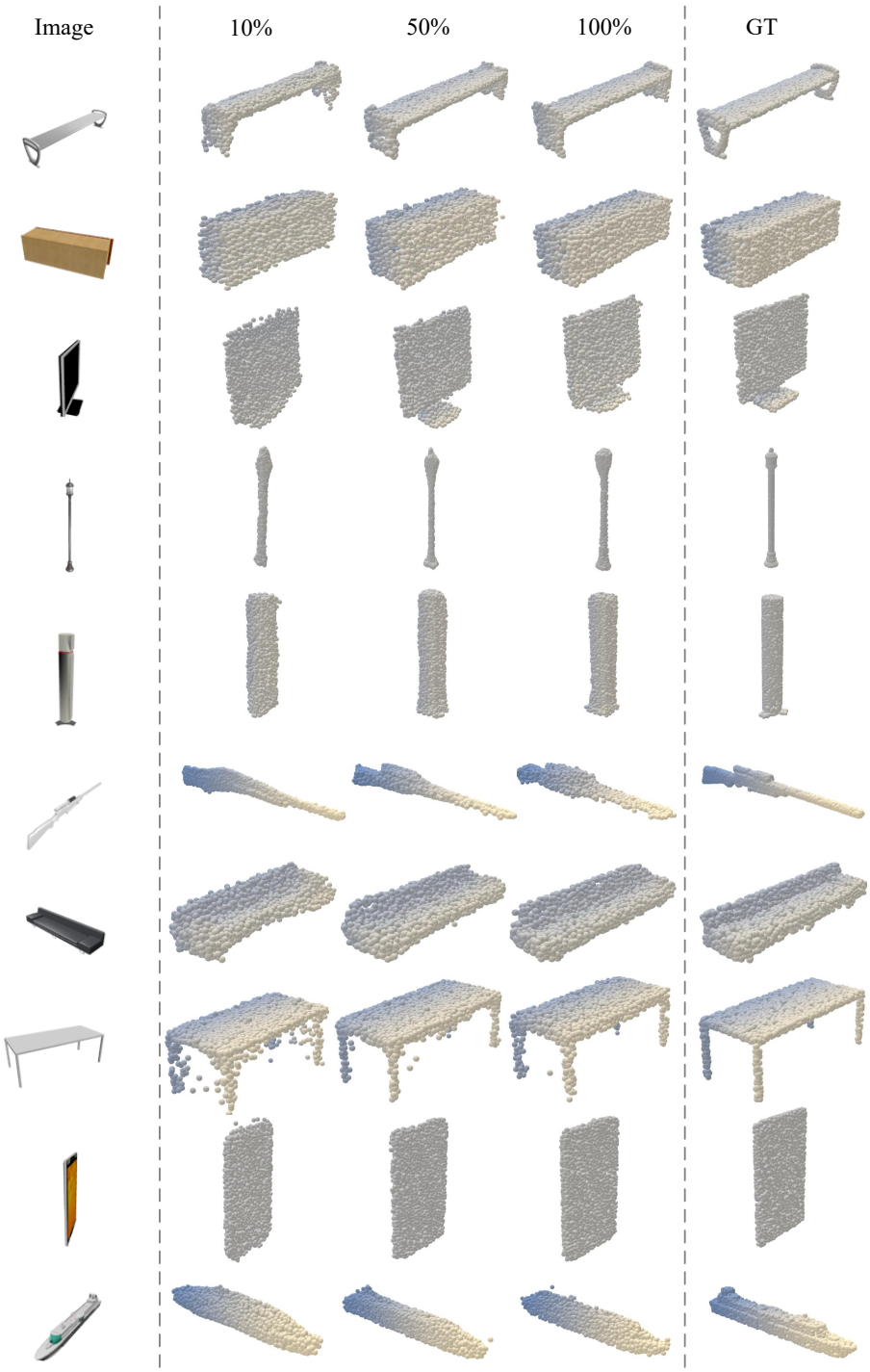


Figure 6: Qualitative reconstruction results of RGF on the remaining ten ShapeNet-R2N2 categories.

(Table 9). The matched PDM comparison supplements the earlier baselines: RGF has lower CD on Chair and Table, higher F1 on all three categories, and trails on Sofa CD. These results retain the original real-scene comparison while making the recent baseline explicit.

Comparison with PC^2 (CVPR 2023): PC^2's implicit conditioning struggles with the diverse backgrounds and variable lighting of real-world images. RGF employs **Geometric Injection (GI)**, which explicitly lifts 2D features to a 3D scaffold derived from monocular depth estimation. This

Table 9: Performance improvement of RGF over Pix3D comparators, including PDM. "-" indicates that RGF is inferior on that metric.

Method	Chair		Sofa		Table	
	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑
PC^2 (CVPR 23) Melas-Kyriazi et al. (2023)	115.94	0.443	47.17	0.445	202.77	0.397
	45.07%	17.16%	8.92%	6.52%	30.87%	9.82%
CCD-3DR (arXiv 23) Di et al. (2023)	111.42	0.456	44.91	0.450	196.28	0.418
	42.84%	13.82%	4.34%	5.33%	28.58%	4.31%
BDM-M (CVPR 24) Xu et al. (2024)	113.40	0.449	44.50	0.451	202.08	0.413
	43.84%	15.59%	3.46%	5.10%	30.63%	5.57%
BDM-B (CVPR 24) Xu et al. (2024)	110.60	0.455	45.05	0.455	186.46	0.429
	42.41%	14.07%	4.64%	4.18%	24.82%	1.63%
MESC-3D (CVPR 25) Li et al. (2025)	91.36	0.370	41.98	0.284	206.16	0.306
	30.29%	40.27%	-	66.90%	32.00%	42.48%
PDM (ECCV 26) Zhou et al. (2026b)	79.28	0.499	**41.43**	0.463	184.35	0.422
	19.66%	4.01%	-	2.38%	23.96%	3.32%
RGF	**63.69**	**0.519**	42.96	**0.474**	**140.18**	**0.436**



Figure 7: Qualitative Pix3D comparison across three real-world object classes.

structural prior bridges the domain gap, resulting in a remarkable 45.07% reduction in CD for the Chair category.

Comparison with BDM-M/BDM-B (CVPR 2024): BDM focuses on unconditional quality but lacks precise semantic alignment mechanisms for single-view reconstruction. RGF's **GCRT** ensures deterministic mapping from noise to data, while GI provides metric guidance. This combination allows RGF to outperform BDM-M with a 43.84% CD reduction in Chair and a 3.46% CD reduction in Sofa.

Additional comparison with MESC-3D (CVPR 2025): MESC-3D can suffer from semantic–geometric misalignment in complex scenes. RGF's **Intrinsic Geometry Aggregation (IGA)** processes features along the object's surface through parallel transport, preserving details such as thin chair legs that may be smoothed by semantic-focused aggregation. In this pairwise reproduction, RGF has higher F1 than MESC-3D by 40.27% on Chair and 66.90% on Sofa. The matched PDM results above provide the current direct comparison on Pix3D.

B.3 Model Efficiency: Parameters and Runtime Optimization

As shown in Table 10, RGF is compared with PC^2 Melas-Kyriazi et al. (2023), BDM Xu et al. (2024), MESC-3D Li et al. (2025), and the recent PDM baseline in terms of efficiency. The PDM comparison is the primary recent efficiency reference, while the earlier rows preserve the historical latency context.

Matched comparison with PDM (ECCV 2026): PDM takes 20.73s/sample and uses 0.39GB of memory, compared with RGF's 1.825s and 3.46GB. Thus, RGF is 11.4× faster at the cost of higher memory use; parameter counts are similar (57.78M vs. 56.34M).

Additional comparison with PC^2/CCD-3DR (CVPR 2023): PC^2 relies on slow iterative diffusion sampling (typically hundreds of steps). RGF's **Geometry-Conditioned Rectified Transport (GCRT)** enables high-fidelity generation in a *single step*. This paradigm shift delivers a 26.6× speedup (1.825s vs 48.46s). Although RGF has slightly more parameters (57.78M vs 47.41M) due to the additional GI/IGA modules, it substantially reduces inference latency.

Additional comparison with BDM-M/BDM-B (CVPR 2024): BDM's complex Bayesian attention mechanisms result in high latency (50s+). RGF's streamlined architecture reduces parameters by

Table 10: Model Efficiency Improvement Percentage (RGF vs. Comparators). "-" indicates that RGF's performance on the corresponding metric is inferior to the comparator.

Method	Parameters (M)	Runtime (s)	GPU Memory (GB)
PC^2/CCD-3DR (CVPR 23)	**47.41**	48.46	1.73
	-	26.6 ×	-
BDM-M (CVPR 24)	74.82	52.47	2.01
	22.77%	28.8 ×	-
BDM-B (CVPR 24)	73.78	50.81	1.93
	21.69%	27.8 ×	-
MESC-3D (CVPR 25)	74.84	34.05	0.86
	22.80%	18.7 ×	-
PDM (ECCV 26) Zhou et al. (2026b)	56.34	20.73	**0.39**
	-	11.4 ×	-
RGF	57.78	**1.825**	3.46

Table 11: Average latency accounting for the online RGF pipeline over 804 samples. Percentages are rounded. Conditioning and point-backbone rows are nested within the one-step forward pass and must not be added again.

Stage	Time (s/sample)	Share
Online Depth Pro lifting	1.683	92.3%
One-step model forward	0.041	2.3%
Conditioning	0.022	1.2%
Point backbone	0.017	0.9%
Export/logging	0.100	5.5%
Full end-to-end	**1.825**	**100.0%**

over 20% and achieves a 28.8× speedup, providing a lower-latency operating point than BDM under the stated hardware and end-to-end protocol.

Additional comparison with MESC-3D (CVPR 2025): MESC-3D employs heavy optimization or sampling processes (34.05s). Under the same reported protocol, RGF takes 1.825s versus 34.05s for MESC-3D (18.7× faster) and uses 22.80% fewer parameters. While RGF utilizes more GPU memory (3.46GB) to support the explicit geometric scaffold (GI) and parallel transport computations, this historical pairwise result is supplementary; PDM is the recent matched reference above.

In summary, the reported advantages arise from the integration of GCRT with the task-specific GI and IGA modules under the aligned point-cloud reconstruction protocols. The resulting accuracy and latency comparisons are scoped to the baselines and hardware for which the input, output, and evaluation procedures were matched.

B.4 Train Curves

The train curves of RGF, PC^2, BDM and MESC-3D are presented in Figure 8, Figure 9, Figure 10 and Figure 11 respectively. Please refer to the ***RGF_PC2_BDM_MESC-3D_train_loss.zip*** file in the appendix for clear images.

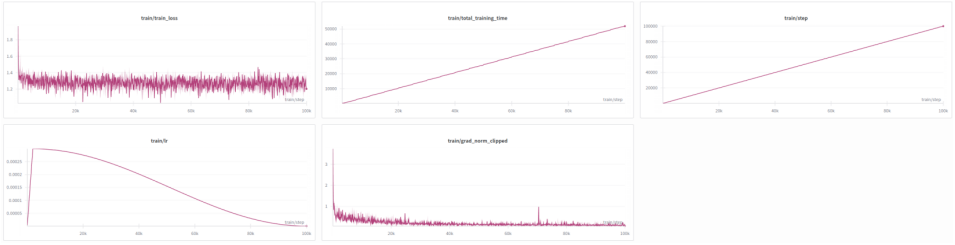


Figure 8: The train curves of RGF.



Figure 9: The train curves of PC^2.

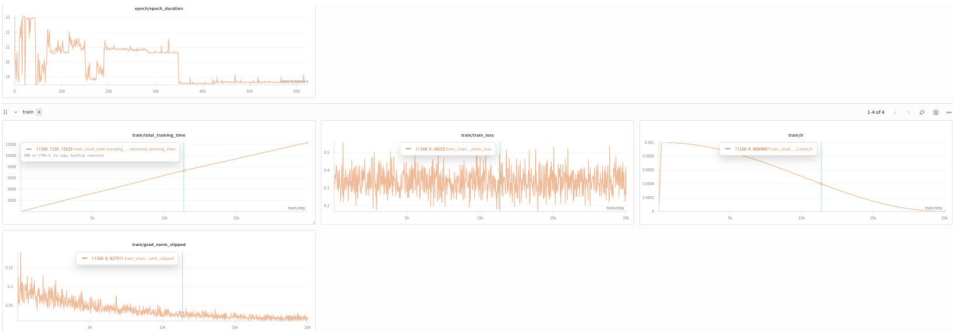


Figure 10: The train curves of BDM.



Figure 11: The train curves of MESC-3D.

C Additional Validation and Clarifications

This section provides additional measurements and formal clarifications that make the empirical and geometric claims self-contained. All claims are stated with their scope: task-specific paired-supervision efficiency is distinguished from the pretraining cost of frozen encoders, and an endpoint-conditioned transport surrogate is distinguished from exact path-integrated parallel transport.

C.1 Evaluation protocol and controlled-analysis scope

All reported reconstruction metrics use the same point-set protocol unless a table states otherwise. Predictions and references contain 4,096 points and are independently centered before evaluation. The implementation computes the bidirectional mean of squared nearest-neighbour distances and reports CD after multiplication by 10^3. The F-score uses the harmonic mean of precision and recall, with the evaluator threshold applied to squared nearest-neighbour distances at 0.01. We retain the conventional "F1@1%" label used by the benchmark tables, but the squared-distance convention is stated here explicitly so that the numerical scale is reproducible.

The controlled analyses are separated by purpose. The Airplane/50% component table removes GI, IGA, or GCRT, whereas the matched-prior table gives PC^2, BDM-B, and MESC-3D access to the same RGF-derived GI features. The latter is a control for prior access, not a claim that the competing architectures have been retrained with an identical fusion module. The scaffold-swap study tests a moderate same-category mismatch, and the NFE study isolates solver-step sensitivity on fixed inputs and noise. These controls support bounded statements about the geometric scaffold and the one-step model; they do not establish robustness to every camera or depth failure mode.

The numerical comparisons are restricted to methods for which the point count, camera protocol, data split, and evaluation code can be aligned. Point-MF and mean-flow generators such as MFM-Point are therefore discussed as methodological context rather than assigned a new number under a reconstruction protocol that they do not natively share. The consistency stage in RGF is measured through the one-step student and its teacher-NFE sweep; an external consistency-distillation implementation is not substituted for the reported RGF pipeline. This distinction keeps direct measurements separate from qualitative positioning.

C.2 Input-prior provenance and implementation conventions

The geometric scaffold used by GI is obtained from a frozen, released Depth Pro checkpoint Bochkovskii et al. (2025). The depth estimator is evaluated on the RGB input and is not fine-tuned on ShapeNet-R2N2 or Pix3D. The ground-truth point clouds in ShapeNet-R2N2 and Pix3D are used for task supervision and evaluation; ground-truth synthetic depth rendered from those point clouds is not supplied to GI. For Pix3D, the cropped image and its recalibrated camera matrix define the evaluation protocol. For ShapeNet-R2N2, the provided rendering and camera metadata define the corresponding projection protocol. The pretraining data and compute of the released Depth Pro checkpoint are external to the task-specific RGF training budget and are therefore not included in the paired image–point supervision count.

The image-conditioning branch uses a frozen ViT-S/16 feature encoder initialized from a self-supervised MSN checkpoint. Thus, the data-efficiency statement concerns the amount of task-specific paired image–point supervision while the external image and depth representations are held fixed. It does not assert that the complete system is independent of representation pretraining.

For the reported GI implementation, scaffold features are interpolated from the three nearest scaffold points after normalized point-set matching. The normalized squared-distance weights use $\mathrm{softmax}(-50d^2)$, and a point is retained when its nearest normalized squared distance is below 0.0025. In the generic interpolation notation, these values correspond to the distance-temperature parameter and the validity threshold, whereas the neighborhood size is $k = 3$. The threshold table in Sec. B.1.1 varies the validity threshold around this operating point; it is not a joint sensitivity study of the distance temperature and neighborhood size. Consequently, the reported results establish the behavior of the fixed protocol and the tested threshold range, but do not imply invariance to arbitrary choices of β, τ, or k.

Depth scale and camera-protocol boundary. For feature matching, the dynamic point state and the depth-derived scaffold are each centered and normalized before the local nearest-neighbor interpolation, so the GI threshold is expressed in normalized matching coordinates rather than in metric depth units. ShapeNet-R2N2 uses the provided rendering camera metadata, while Pix3D uses the cropped image and its recalibrated camera matrix. The current protocol therefore supports bounded behavior for the stated normalization and camera conventions; it does not establish robustness to oracle depth, a different depth estimator, depth noise, focal-length or pose errors, or crop misalignment.

C.3 Efficiency accounting and speedup definitions

End-to-end speedup definition. The measured end-to-end RGF time is 1.825 seconds per sample, including online depth lifting. The primary matched comparison is PDM at 20.73 seconds/sample, corresponding to an $11.4\times$ speedup. For completeness, ratios against earlier reported baselines are $26.6\times$ for PC^2/CCD-3DR (48.46s), $28.8\times$ for BDM-M (52.47s), $27.8\times$ for BDM-B (50.81s), and $18.7\times$ for MESC-3D (34.05s). Depth lifting dominates RGF's online cost at 1.683 seconds (92.3%). The one-step forward pass takes 0.041 seconds (2.3%); conditioning and point-backbone measurements (0.022 and 0.017 seconds) are nested sub-measurements, not additional stages. Export and logging take 0.100 seconds (5.5%), yielding the 1.825-second end-to-end total after rounding. The current measurements do not provide an independent GI/IGA/decoder decomposition.

Geometric formulation. The complete definitions and proofs are given in Sec. H. In brief, the learned split $\mathbf{h}_i \mapsto (\mathbf{b}_i, \boldsymbol{\gamma}_i)$ is a local trivialization, and the skew-symmetric exponential gives an exact $SO(d_f)$ transport. The curvature-inspired branch is a learned finite-dimensional residual map on the base coordinates; it is not presented as recovery of a physical connection or as automatic differentiation of a Levi–Civita field. Lemma H.1 proves norm and inner-product preservation; Theorem H.2 gives the frame-covariant attention result; Proposition H.3 bounds the perturbation of the pre-softmax score; and the residual-stability statement in Sec. H.3 gives the corresponding conditional bound. We explicitly do not claim exact holonomy or path composition. The implemented object is an endpoint-conditioned isometric transport surrogate inspired by parallel transport.

C.4 Single-step quality, stability, and baselines

Matched PDM comparison and additional MESC-3D result. Against PDM, RGF reduces Chair CD at 10% data from 82.41 to 76.39 (7.30%). Across the nine ShapeNet-R2N2 category/data settings, RGF lowers CD in seven and raises F1 in six, with one F1 tie. On Pix3D, it lowers CD in two of three categories and raises F1 in all three (Table 14). These matched results replace the earlier headline comparison as the primary evidence. A separate Chair/10% comparison with MESC-3D gives:

$$\frac{101.51 - 76.39}{101.51} = 24.75\%. \tag{22}$$

The 24.75% value is specific to this MESC-3D pairwise result, not an all-category average or a comparison with the best current baseline. For reference, the reduction relative to the best Chair/10% baseline in the historical table (CCD-3DR, CD $= 89.79$) is 14.92%, while the all-category 10% average in Table 2 compares RGF (75.98) with BDM (93.87), a 19.1% reduction. The PDM values above are the aligned recent comparison used for the main claims.

Multi-seed stability. The original submission used seed 42. We subsequently repeated the three principal ShapeNet-R2N2 categories with seeds 42, 3407, and 114514. The entries below are mean $\pm$ sample standard deviation; each cell reports CD/F1.

Table 12: Three-seed stability on ShapeNet-R2N2. The study covers the three categories used in the main comparison; it quantifies variance and is not a formal significance test.

Category	1%	10%	50%	100%
Airplane	$84.30 \pm 0.82/0.581 \pm 0.009$	$57.29 \pm 0.46/0.670 \pm 0.007$	$44.73 \pm 0.35/0.712 \pm 0.006$	$40.58 \pm 0.27/0.714 \pm 0.006$
Car	$62.06 \pm 0.54/0.496 \pm 0.009$	$56.03 \pm 0.43/0.548 \pm 0.005$	$51.51 \pm 0.36/0.582 \pm 0.006$	$53.08 \pm 0.37/0.579 \pm 0.004$
Chair	$120.36 \pm 1.11/0.352 \pm 0.010$	$76.37 \pm 0.46/0.436 \pm 0.003$	$58.45 \pm 0.27/0.479 \pm 0.005$	$56.23 \pm 0.26/0.465 \pm 0.004$

The CD standard deviation is 0.26–1.11 (less than 1.0% of the corresponding mean), and the F1 standard deviation is 0.003–0.010. Thus the reported low-data trend is stable for these categories, while broader seed coverage remains a useful extension.

Statistical and ablation scope. The three-seed check covers Airplane, Car, and Chair only and is intended to describe run-to-run variation rather than to provide a formal significance analysis. The all-category tables report the main model under the stated training settings; they are not component-wise ablations over every category. The controlled component ablation is confined to the Airplane/50% protocol, so its module-level differences should not be extrapolated as dataset-wide effects.

Single-step quality. We evaluated the same 100 objects, RGB inputs, and noise realizations with the rectified teacher at several numbers of function evaluations (NFE) and with the distilled student. "Model latency" excludes the one-time depth scaffold, whereas "sampler latency" includes the solver call for the reported fixed scaffold.

Table 13: NFE ablation for the rectified teacher and one-step student.

Model	NFE	CD ↓	F1 ↑	Model / sampler latency (s)
Teacher	10	51.32	0.657	0.2227 / 0.2854
Teacher	20	51.23	0.659	0.3614 / 0.4223
Teacher	50	**51.22**	0.659	0.8461 / 0.8948
Teacher	1,000	51.24	0.659	16.3797 / 16.4317
Student	1	52.63	**0.665**	**0.0545 / 0.1210**

Teacher quality saturates by 20–50 NFE. Relative to the 1,000-NFE teacher, the one-step student's CD increases by only 2.71% and its F1 is slightly higher, while model latency is reduced by 300.3×. The full online path remains 1.825 seconds because it also includes Depth Pro (1.683 seconds).

Sampling and multimodality boundary. With the image, scaffold, checkpoint, and initial noise fixed, the one-step student defines a deterministic output. Resampling the initial noise can produce multiple candidate reconstructions, but this mechanism is not a calibrated uncertainty estimate. The current protocol fixes the noise for the NFE comparison and does not report a diversity–quality curve or uncertainty calibration; consequently, multimodal coverage of the single-view posterior is not established.

Recent and deterministic baselines. We reproduced the newly accepted ECCV 2026 Point Diffusion Mamba (PDM) Zhou et al. (2026b) under the same point count, view protocol, and data ratios. The compact cells list CD/F1 for Chair, Airplane, and Car in that order (and Chair, Sofa, and Table for Pix3D).

RGF lowers CD in nine of twelve direct cells (seven of nine ShapeNet cells and two of three Pix3D cells). It has higher F1 in nine cells, ties PDM on Chair at 100% data, and is lower on only the Airplane and Car F1 values at 100%; PDM is also better on Car CD at 10% and 100% and on Sofa CD. PDM uses 20.73 seconds/sample and 0.39 GB inference memory; RGF uses 1.825 seconds/sample and 3.46 GB, so the speed–memory trade-off is part of the comparison.

We also evaluated a deterministic non-diffusion 3D Attribute Flow regressor on the matched ShapeNet-R2N2 10% protocol. The entries are CD/F1/exact uniform-mass W_1 EMD $\times 100$.

RGF wins all nine comparisons and lowers the sample-weighted EMD from 63.64 to 30.23 (52.5%). For completeness, the exact EMD comparison on the 6,279-object common cohort is given in Table 16; all clouds are independently centered and contain 4,096 points.

Metric scope beyond CD and F1. The supplementary EMD analyses use exact uniform-mass W_1 distances and provide a transport-based complement to the bidirectional squared-distance CD and thresholded F1. A point-cloud prediction in the reported protocol contains positions but no consistently defined surface-normal field. Per-point normal accuracy would therefore require an additional surface reconstruction or a prescribed normal-estimation and correspondence procedure, neither of which is part of the current evaluator. The CD, F1, and EMD values are consequently interpreted as point-position and mass-transport measures; no claim about normal accuracy is inferred from them.

Table 14: Comparison with the recent PDM baseline Zhou et al. (2026b). RGF results are from the corresponding entries in the main tables; bold marks the lower CD or higher F1 when the comparison is direct.

Setting	PDM CD/F1	RGF CD/F1
R2N2 10%	82.41/0.419; 60.64/0.614; 55.23/0.542	**76.39/0.437; 57.30/0.669; 56.00/0.547**
R2N2 50%	68.85/0.465; 50.14/0.663; 55.53/0.558	**58.46/0.478; 44.71/0.711; 51.49/0.581**
R2N2 100%	62.14/**0.488**; 48.66/**0.719**; **51.54/0.607**	**56.24/0.488**; **40.60**/0.713; 53.12/0.578
Pix3D	79.28/0.499; **41.43**/0.463; 184.35/0.422	**63.69/0.519**; 42.96/**0.474**; **140.18/0.436**

Table 15: Deterministic non-diffusion comparison at 10% training data.

Category (common EMD N)	3D Attribute Flow	RGF
Airplane (808)	115.75/0.564/66.70	**57.30/0.669/31.89**
Car (707)	108.55/0.384/45.65	**56.00/0.547/25.15**
Chair (1,323)	124.74/0.283/71.38	**76.39/0.437/31.94**

C.5 Generalization and recent comparisons

To probe distribution shift beyond the canonical ShapeNet-R2N2 and Pix3D protocols, we evaluated RGF on screened Objaverse assets and on real captures from CO3Dv2. These experiments are adaptation studies rather than unrestricted zero-shot claims. All point clouds contain 4,096 points and are evaluated with the same independently centered CD and F1@1% implementation used for the main experiments.

For Objaverse, 662 assets were manually screened to remove invalid geometry, unsuitable topology, and unusable views. Standardized RGB views and camera parameters were rendered in Blender. RGF and BDM-B use the full corresponding training split, 20,000 training steps, and seed 42; MESC-3D uses the same training ratio, point count, and seed for 300 epochs. CD is reported in 10^{-3}.

RGF is best in all six category–metric comparisons. Relative to BDM-B, CD is reduced by 26.2%, 8.9%, and 4.2% on Airplane, Car, and Chair, respectively; relative to MESC-3D, the corresponding reductions are 29.5%, 16.8%, and 9.4%. The F1 gains relative to BDM-B are 0.031, 0.020, and 0.005, and those relative to MESC-3D are 0.040, 0.033, and 0.013.

We additionally used the processed CO3Dv2 "fewview_dev" split for Hydrant and Teddybear. The setting uses an object-level 10% training split, the complete test split, masked 224×224 inputs, seed 42, and independently centered CD/F1/exact-EMD evaluators. CD and EMD are reported in 10^{-3} and 10^{-2} units, respectively. RGF and BDM-B train for 10,000 steps; MESC-3D trains for 200 epochs with frozen prompt and text encoders.

RGF wins all six CO3Dv2 comparisons. Relative to BDM-B, CD falls by 3.3% and 2.0%, F1 rises by 0.004 and 0.005, and EMD falls by 1.6% and 0.9% on Hydrant and Teddybear. Relative to MESC-3D, the CD reductions are 38.4% and 23.6%, the F1 gains are 0.187 and 0.109, and the EMD reductions are 56.7% and 44.9%.

The qualitative evidence is organized by diagnostic purpose. Figures 20–25 expose the RGB-to-depth-to-scaffold construction; Fig. 5 covers the extreme low-data cases; and Fig. 6 contains the remaining

Table 16: Exact uniform-mass W_1 EMD $\times 100$ (mean $\pm$ sample SD) on the common 100%-data cohort. Lower is better; bold marks the best entry in each row.

Category (N)	BDM-B	MESC-3D	RGF
Airplane (416)	31.55±15.41	89.60±14.94	**27.10±11.70**
Bench (361)	**20.93±9.72**	52.81±13.87	21.59±9.88
Cabinet (312)	25.08±7.02	79.30±9.36	**22.74±5.54**
Car (448)	25.60±13.06	47.40±6.91	**23.46±9.53**
Chair (448)	27.58±8.13	82.58±9.74	**27.29±7.13**
Display (218)	27.31±14.39	91.43±11.23	**25.91±12.33**
Lamp (463)	39.40±23.25	114.06±29.70	**36.32±21.07**
Loudspeaker (320)	29.27±10.54	81.36±15.71	**27.33±8.87**
Rifle (475)	**24.02±8.74**	83.12±15.55	26.54±8.22
Sofa (630)	23.05±6.26	72.26±11.00	**22.45±7.22**
Table (1,685)	27.26±11.17	71.54±17.32	**27.05±11.07**
Telephone (115)	19.54±11.83	61.70±10.27	**19.45±11.45**
Watercraft (388)	24.63±8.65	87.57±12.77	**23.71±8.43**
Pooled (6,279)	27.00±12.74	77.19±22.09	**26.10±11.45**

Table 17: Objaverse generalization on standardized renders.

Method	Airplane CD/F1	Car CD/F1	Chair CD/F1
RGF	**5.53/0.923**	**10.78/0.846**	**14.00/0.841**
BDM-B	7.49/0.892	11.83/0.826	14.61/0.836
MESC-3D	7.84/0.883	12.95/0.813	15.45/0.828

Table 18: CO3Dv2 real-capture adaptation results. EMD is exact uniform-mass W_1.

Method	Hydrant CD/F1/EMD	Teddybear CD/F1/EMD
RGF	**156.77/0.335/42.14**	**207.31/0.251/53.35**
BDM-B	162.16/0.331/42.83	211.57/0.246/53.82
MESC-3D	254.67/0.148/97.39	271.41/0.142/96.75

ShapeNet categories. These panels are used for visual inspection only when no paired 3D ground truth is available; web-image stress cases are not used to claim a numerical improvement.

The PDM comparison in Table 14 covers the recent single-view point-cloud setting under matched point count, camera protocol, and data ratios. RGF lowers CD in nine of twelve cells and has higher F1 in nine cells, with a tie on Chair at 100% data; PDM is better on Car CD at 10%/100%, Sofa CD, and Airplane/Car F1 at 100%. Both metrics are therefore reported rather than collapsed into one ranking. The measured PDM inference time is 20.73 seconds/sample with 0.39 GB memory, compared with RGF's 1.825 seconds/sample and 3.46 GB.

Metadata consistency. The title used by the source, PDF title page, and submission metadata is "Single-Step 3D Reconstruction via Geometry-Conditioned Rectified Transport." Keeping this string identical avoids ambiguity between the method name and the task description.

C.6 Compute, depth robustness, and data efficiency

The 1,000-step Euler integration used for rectification is an offline, gradient-free teacher rollout. It is not unrolled through the gradient graph of every training update: conditional flow matching samples one time t, constructs one interpolated state, and evaluates the velocity once. The measured cost for all 13 ShapeNet-R2N2 categories is shown below.

Table 19: Four-stage training and teacher-generation cost on one RTX 3090.

Stage	GPU-hours
Velocity/CFM training	16.869
Offline pair generation (1,000-step, `no_grad`)	13.110
Rectified-flow training	15.481
Consistency training	15.865
Gradient subtotal	48.215
Complete four-stage total	**61.325**

For 3,096 unique training objects, the gradient stages cost 56.06 GPU-seconds per sample and all four stages cost 71.31 GPU-seconds per sample (2.555 GPU-days total). The one-time offline cost is amortized over repeated inference calls. Batch-size-one profiles, measured after ten warm-up and fifty timed iterations with loading and I/O excluded, are:

Because no 1,000-step graph is differentiated, distributed data-parallel training stores one model replica and local-batch activations per rank, with the usual one gradient all-reduce per update. The activation memory and synchronization frequency therefore do not acquire a 1,000-fold factor.

To quantify sensitivity to an incorrect geometric scaffold, we performed a controlled same-category cross-object swap on 12 fixed targets from each of Airplane, Car, and Chair. RGB, checkpoint, noise, point count, and evaluator were held fixed. Each own scaffold was replaced by a moderately mismatched same-category object selected deterministically from 64 candidates while excluding extreme scale ratios.

The mean CD increase is 0.82% and the F1 change is -0.0006. This supports a bounded statement: image and transport branches prevent a moderately wrong same-category scaffold from uniquely determining the output, but severe depth or camera failures can still bias the reconstruction.

Table 20: Per-sample training-step profiles and peak memory.

Stage	Full (ms)	Forward (ms)	Backward (ms)	Optimizer (ms)	Peak (GiB)
RGF velocity/CFM	74.411	37.832	22.030	14.549	3.978
RGF rectified flow	85.392	46.446	23.228	15.718	3.978
RGF consistency	75.157	39.266	21.648	14.244	3.978
BDM diffusion	161.604	80.760	65.545	15.298	7.282

Table 21: Robustness to a moderately mismatched depth scaffold.

Category	Own scaffold CD/F1	Swapped scaffold CD/F1	Delta CD/F1
Airplane	60.05/0.6725	60.24/0.6718	+0.19/-0.0007
Car	135.75/0.4341	135.61/0.4347	-0.13/+0.0005
Chair	151.38/0.3311	154.18/0.3295	+2.79/-0.0016
36-object mean	115.73/0.4792	116.68/0.4786	+0.95/-0.0006

The task-specific data-efficiency claim is separated from external pretraining. The image and depth encoders remain frozen while the amount of paired image–point supervision changes. The internal Airplane/50% ablation is:

Table 22: Component ablation under the Airplane/50% protocol (CD/F1).

Variant	CD ↓	F1 ↑
Full RGF	**44.71**	**0.711**
Without GI	48.59	0.632
Without IGA	47.75	0.676
Without GCRT	72.23	0.618

We also supplied the same RGF GI features to PC^2, BDM-B, and MESC-3D, using Airplane/50% checkpoints and all 809 test objects:

The scaffold helps PC^2 but not BDM-B or MESC-3D, while restoring GI inside RGF improves both metrics. This indicates that the gain is not a plug-and-play depth-prior effect; it depends on the integration of GI with IGA and GCRT. The precise claim is therefore "data-efficient with respect to task-specific paired 2D–3D supervision under fixed pretrained image and depth encoders," rather than end-to-end efficiency independent of pretraining. Frozen pretrained representations are also used by the compared methods (for example, CLIP- or PointBERT-based encoders), although RGF additionally uses Depth Pro; this additional prior is disclosed explicitly.

C.7 Controlled-analysis boundaries and open diagnostics

The Airplane/50% component ablation and matched-prior table are fixed-protocol controls over all 809 Airplane test objects, with the same camera protocol, point count, evaluator, and data ratio. They isolate the contribution of the modules and the effect of exposing competing architectures to the same RGF-derived geometric features within this setting. They should not be read as a dataset-wide equalization result: the current appendix does not provide a corresponding GI-on/GI-off comparison on Pix3D or a matched-depth-prior rerun for every category and dataset.

The scaffold-swap result is deliberately limited to a moderate same-category mismatch with extreme scale ratios excluded. It supports tolerance to a bounded scaffold error, but it does not establish invariance to camera intrinsics or extrinsics perturbations, severe depth noise, missing depth, or adversarial outliers. Those stress regimes require separate corruption protocols because they change the input geometry rather than only replacing it with a nearby scaffold.

The full IGA ablation measures the contribution of the implemented module as a whole. It does not by itself identify the separate effects of the curvature-inspired residual, the $SO(d_f)$ parameterization, or the additional feature capacity. Curvature-only and parameter-matched variants, explicit removal of the orthogonal constraint, and tokenization-sensitivity sweeps are therefore treated as open diagnostics rather than as completed results. Equation 36 defines the appropriate finite-precision diagnostic for $\mathbf{P}_{ij}$; no empirical error distribution is implied here without a recorded instrumentation run.

Finally, the latency table reports end-to-end timing together with nested forward-pass measurements. It does not constitute an independent GI/IGA/decoder profile. Such a profile would require synchronized module-level events and a fixed policy for accounting for shared feature extraction, memory

Table 23: Matched depth-prior access for competing architectures (CD/F1).

Method	Native	With RGF GI/fusion
PC^2	76.39/0.628	69.92/0.635
BDM-B	68.66/0.641	69.58/0.637
MESC-3D	51.28/0.619	62.71/0.399
RGF (without GI / full)	48.59/0.632	**44.71/0.711**

movement, and kernel warm-up. These boundaries keep the reported evidence reproducible while identifying the measurements needed for a more granular deployment analysis.

The reported 3.46 GiB is the peak allocated memory of the batch-size-one end-to-end inference path under the stated runtime protocol; it is not the sum of independent GI, IGA, tokenization, and decoder peaks. The current measurements therefore do not assign an exact memory share to any module. In general, local transport storage can grow with the number of tokens, neighbors, and d_f^2 matrix entries. Mixed precision, activation checkpointing, or lower-rank skew parameterizations are possible engineering directions, but their memory savings are not measured here.

C.8 Cross-representation comparison scope

Single-view reconstruction systems can produce point clouds, meshes, triplanes, or Gaussian representations, and their reported latency may include different image encoders, preprocessing, representation conversion, and rendering stages. The present protocol evaluates a 4,096-point reconstruction together with online depth lifting and point-set metrics. Feed-forward systems based on transformer, Gaussian, or triplane representations therefore provide useful architectural context, but their published latency and quality values are not direct substitutes for a matched point-cloud comparison unless the input resolution, camera metadata, output representation, point count, preprocessing, hardware, and evaluator are aligned.

The same boundary applies to consistency-distilled and rectified-flow methods. RGF's consistency phase is an internal teacher–student training stage, whereas an external consistency baseline would require its own implementation, checkpoint, and reconstruction-specific conditioning. The teacher NFE sweep, the matched PDM comparison, and the deterministic Attribute Flow control characterize the operating points measured in this work; they are not presented as a complete ranking over all feed-forward, consistency, Gaussian, mesh, or one-step flow systems. The efficiency statement is therefore restricted to the aligned reconstruction baselines and deterministic controls reported in this appendix.

C.9 Scope, limitations, and deployment considerations

RGF is designed as an explicit point-cloud reconstruction module, not as a complete open-world asset-generation system. The 4,096-point output is the common ShapeNet-R2N2 and Pix3D protocol inherited by the baselines; higher point densities and downstream mesh or texture reconstruction are natural extensions but are not claimed as demonstrated results here. In particular, no experiment in this appendix feeds RGF points into a commercial meshing or texturing pipeline, so statements about downstream initialization should be read as a proposed integration path rather than an evaluated benefit.

The combined evidence supports four operational boundaries: (i) offline teacher-pair generation is a one-time cost; (ii) Depth Pro dominates the online latency; (iii) soft fusion provides bounded tolerance to moderate scaffold errors, not immunity to catastrophic depth or camera failures; and (iv) task-specific data efficiency does not erase the compute and data used to pretrain frozen encoders. The PDM comparison further shows a speed–memory trade-off: RGF is substantially faster but uses more inference memory. These boundaries motivate future work on uncertainty-aware depth fusion, higher-density outputs, downstream meshing/texturing evaluation, and open-world real-capture benchmarks.

RGF is best in 11 of 13 categories and lowers pooled EMD by 3.34% versus BDM-B and 66.19% versus MESC-3D. Unequal-cohort pooled means (RGF 26.26 on $N = 7,806$, BDM-B 27.00 on $N = 6,279$, MESC-3D 77.36 on $N = 7,806$) are not used for ranking.

D ADDITIONAL MESC-3D REPRODUCTION NOTES

MESC-3D Li et al. (2025) is a related earlier method with reported comparisons to BDM Xu et al. (2024) and PC^2 Melas-Kyriazi et al. (2023). To ensure a fair historical comparison, we attempted to reproduce MESC-3D from the public GitHub repository and recovered the missing components and dependencies needed for evaluation. This section documents that reproduction and its pairwise results as supplementary context; the main paper uses the matched recent PDM comparison as its primary diffusion baseline.

D.1 PRACTICAL REPRODUCTION ISSUES

Pre-trained model availability The repository lists several pre-trained models (e.g., BaseModel, ProModel, ULIPModel) required for training and inference (Figure 12). In our reproduction attempt, these files were not available in the local repository (Figure 13), and the provided Google Drive links returned access restrictions:

- Access permission was required to view the linked content (Figure 14).
- We did not receive access during the reproduction period (Figure 14).

We therefore recovered the required components from publicly available resources and used them in the reproduced setup.

```
ckpt_path: Path/ckpt_path/pro_model.pth
prompt_ckpt_path: Path/prompt.pth
text_encoder_ckpt_path: Path/text_encoder.pth
ckpts: checkpoints/modelnet_1k.pth
```

Figure 12: The .pth files required in MESC-3D.

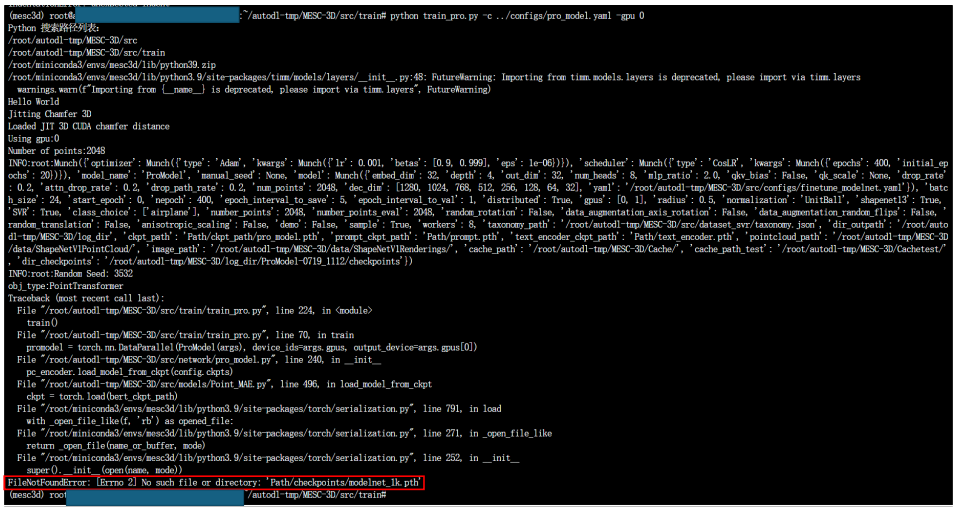


Figure 13: The released MESC-3D code reports missing .pth files in our reproduction environment.

Training and testing scripts The public GitHub repository contained partial code at the time of our reproduction. The README.md states: *"The remaining code is on the way."*. Some pipeline components and paths needed for our experimental protocol therefore had to be completed or adapted before evaluation (Figure 15, Figure 16, and Figure 17).

D.2 REPRODUCTION SETUP

After recovering the missing components, we configured MESC-3D under the same evaluation protocol used for our comparisons, following the BDM Xu et al. (2024) setting where applicable. Parts of the reproduced MESC-3D train and test logs are shown in Figure 18 and Figure 19; the complete logs are included in ***MESC-3D_train_logs.zip*** and ***MESC-3D_test_logs.zip***. Quantitative comparisons with MESC-3D are reported in Table 24, Table 25, and Table 26.

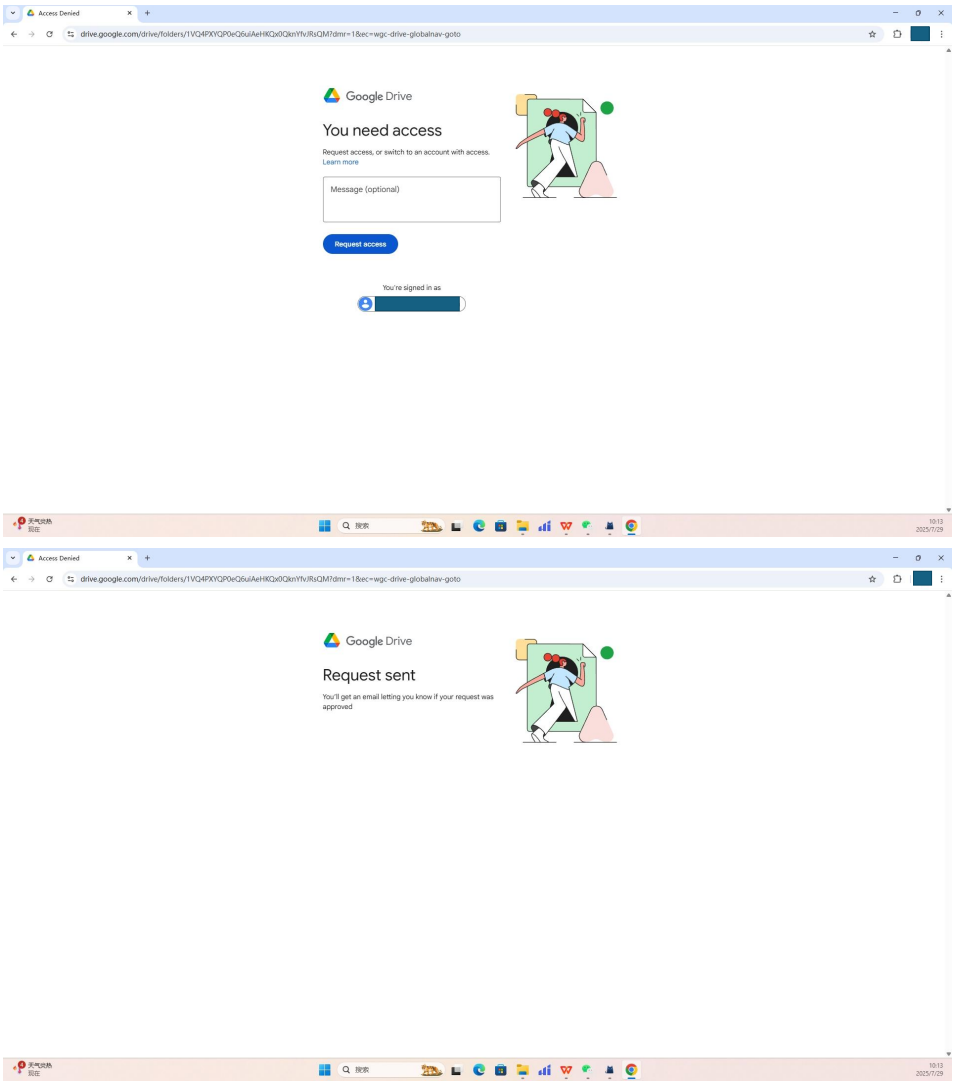


Figure 14: The linked Google Drive resources required access permission during our reproduction attempt.

ShapeNet-R2N2 Dataset: Single-View 3D Reconstruction Performance and Low-Data Robustness As shown in Table 24, on the ShapeNet-R2N2 dataset, RGF demonstrates significant pairwise advantages over the reproduced MESC-3D comparator in both Chamfer Distance (CD, lower is better) and F1 score (higher is better), with more prominent improvements in the low-data regime (10% training data). Taking the Chair category as an example, CD decreases by 24.75% and F1 increases by 14.70% under 10% data; for the Airplane category, CD decreases by 22.89% under 10% data; for the Car category, F1 increases by 4.79% under 10% data. These values describe this reproduced comparator; matched results against PDM are reported in Table 14.

RGF combines deterministic ODE transport with a depth-derived scaffold and parallel-transport feature alignment. These components explain the historical MESC-3D pairwise gains, while the PDM comparison separately evaluates the same design against a recent diffusion-state-space reconstruction method.

Table 24: Performance improvement percentage (RGF vs. MESC-3D) on three training data scales (10%, 50%, and 100%) of ShapeNet-R2N2.

method	Chair						Airplane						Car					
	10%		50%		100%		10%		50%		100%		10%		50%		100%	
	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑
MESC-3D (CVPR 25)	101.51	0.381	73.47	0.427	65.69	0.458	74.31	0.611	51.28	0.619	50.54	**0.714**	56.28	0.522	**44.48**	0.532	**51.99**	**0.602**
RGF	**76.39**	**0.437**	**58.46**	**0.478**	**56.24**	**0.488**	**57.30**	**0.669**	**44.71**	**0.711**	**40.60**	0.713	**56.00**	**0.547**	51.49	**0.581**	53.12	0.578
RGF vs. MESC-3D	+24.75%	+14.70%	+20.43%	+11.94%	+14.39%	+6.55%	+22.89%	+9.49%	+12.81%	+14.86%	+19.67%	-	+0.50%	+4.79%	-	+9.21%	-	-

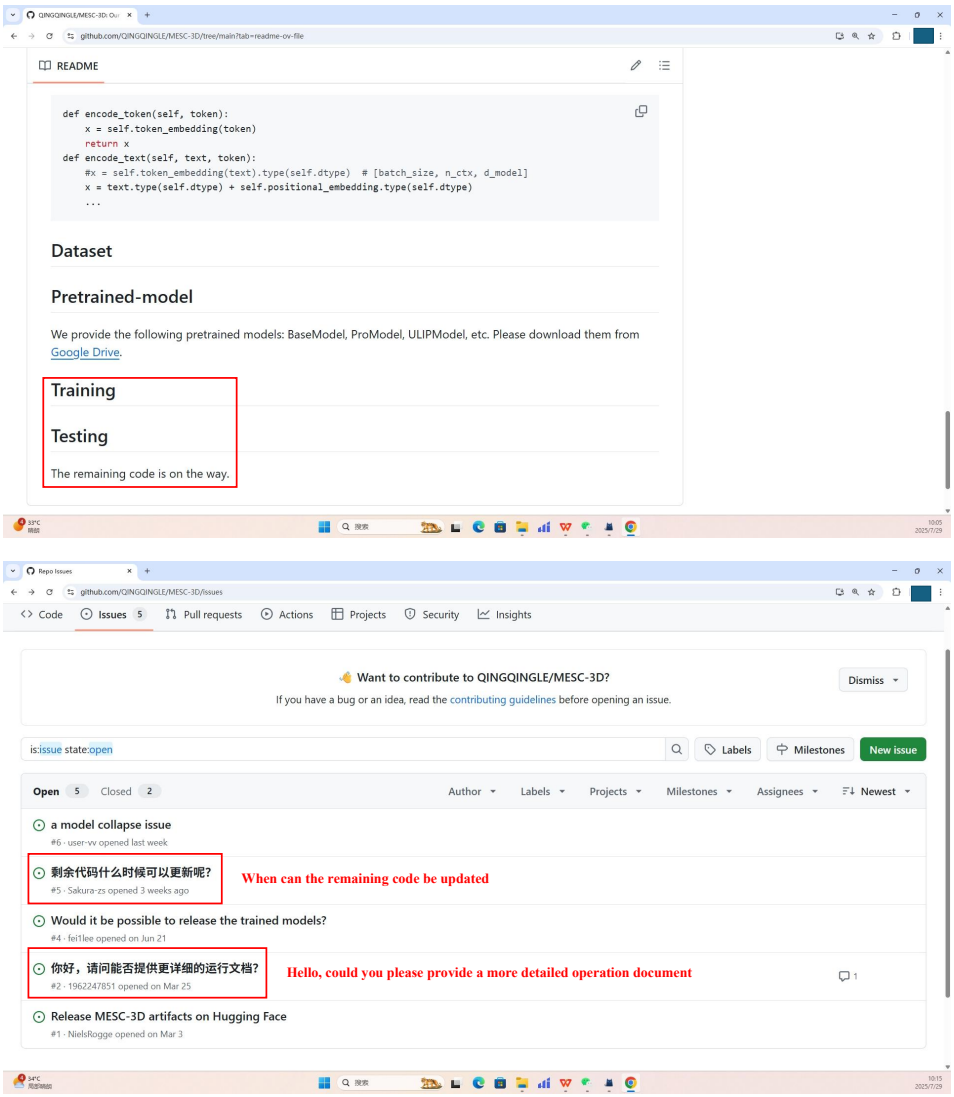


Figure 15: The public repository did not include all training and testing details required by our reproduction protocol.

Pix3D Dataset: Generalization Ability in Real-Scene Single-View 3D Reconstruction As shown in Table 25, on the real-scene Pix3D dataset, RGF improves the reproduced MESC-3D comparison in geometric accuracy (CD) and semantic consistency (F1), especially for F1. For instance, in the Chair category, CD decreases by 30.29% and F1 increases by 40.27%; in the Sofa category, F1 increases by 66.90%; in the Table category, CD decreases by 32.00% while F1 increases by 42.48%. The matched PDM comparison is summarized in Table 14.

RGF's superior pairwise behavior stems from its GI module, which fuses high-level image semantics with an explicit 3D depth scaffold and addresses image–semantic/3D–geometric alignment. Moreover, IGA processes features along the manifold surface via parallel transport, preserving intricate structural details that can be smoothed by semantic-focused aggregation. The paired MESC-3D results above are supplementary to the PDM benchmark in the main comparison.

Table 25: Performance improvement percentage (RGF vs. MESC-3D) on Pix3D.

Method	Chair		Sofa		Table	
	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑
MESC-3D (CVPR 25) Li et al. (2025)	91.36	0.370	**41.98**	0.284	206.16	0.306
RGF	**63.69**	**0.519**	42.96	**0.474**	**140.18**	**0.436**
RGF vs. MESC-3D	+30.29%	+40.27%	-	+66.90%	+32.00%	+42.48%

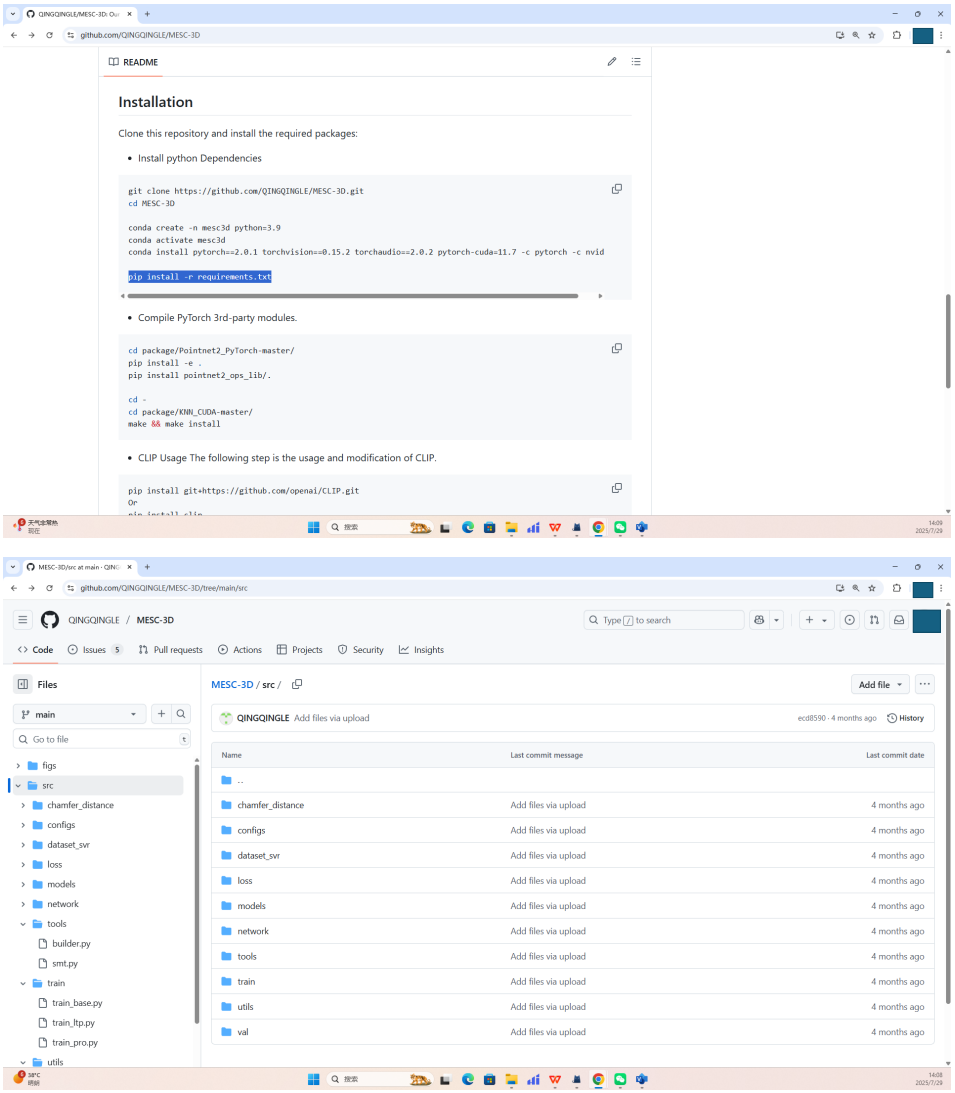


Figure 16: The requirements.txt file was not available in the checked repository snapshot.

```
cd package/Pointnet2_PyTorch-master/
pip install -e .
pip install pointnet2_ops_lib/.

cd -
cd package/KNN_CUDA-master/
make && make install
```

Figure 17: Some documented paths were not available in the checked repository snapshot.

Model Efficiency: Inference Speed and Parameter Efficiency As shown in Table 26, RGF improves computational efficiency relative to the reproduced MESC-3D comparator: it uses 22.80% fewer parameters and is 18.7× faster under the reported measurements. The primary recent comparison is PDM, for which RGF is 11.4× faster with higher memory use (Table 10); the MESC-3D result is retained as supplementary historical context.

This efficiency stems from the GCRT paradigm. Unlike diffusion or optimization-based methods that require iterative sampling steps, RGF learns straight-line ODE trajectories, enabling high-fidelity 3D reconstruction in a deterministic single step. Furthermore, by employing lightweight IGA instead of

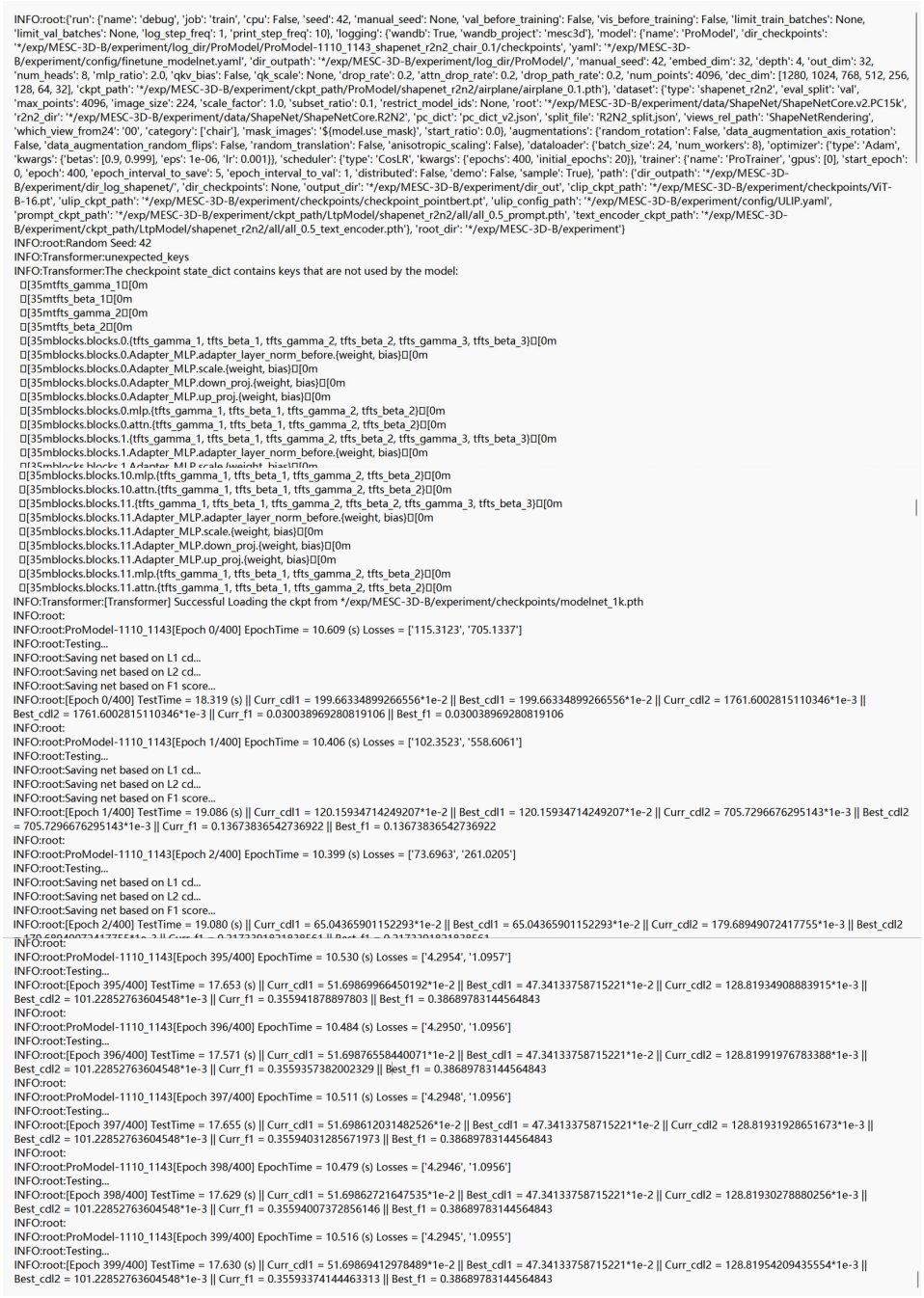


Figure 18: The train logs of MESC-3D on 10% ShapeNet-R2N2.

heavy voxel-based backbones, RGF maintains a compact parameter footprint without compromising reconstruction quality.

Table 26: Performance improvement percentage (RGF vs. MESC-3D) on model efficiency.

Method	Parameters (M) ↓	Runtime (s) ↓	GPU Memory (GB) ↓
MESC-3D (CVPR 25)	74.84	34.05	**0.86**
RGF	**57.78**	**1.825**	3.46
RGF vs. MESC-3D	+22.80%	+18.7 ×	-

2025-11-10 17:06:46,545 - INFO - Configuration: {'run': {'name': 'debug', 'job': 'sample', 'cpu': False, 'seed': 42, 'manual_seed': None, 'val_before_training': False, 'vis_before_training': False, 'limit_train_batches': None, 'limit_val_batches': None, 'log_step_freq': 1, 'print_step_freq': 10}, 'logging': {'wandb': True, 'wandb_project': 'mesc3d'}, 'model': {'name': 'chair_0.1', 'dir_checkpoints': None, 'yaml': '[PROJECT_ROOT]/MESC-3D-B/experiment/config/finetune_modelnet.yaml', 'dir_outpath': '[PROJECT_ROOT]/MESC-3D-B/experiment/log_dir/ProModel/', 'manual_seed': 42, 'embed_dim': 32, 'depth': 4, 'out_dim': 32, 'num_heads': 8, 'mlp_ratio': 2.0, 'qkv_bias': False, 'qk_scale': None, 'drop_rate': 0.2, 'attn_drop_rate': 0.2, 'drop_path_rate': 0.2, 'num_points': 4096, 'dec_dim': [1280, 1024, 768, 512, 256, 128, 64, 32], 'ckpt_path': '[PROJECT_ROOT]/MESC-3D-B/experiment/ckpt_path/ProModel/shapenet_r2n2/GS_BDMNormalization_notTanh_lr_0.001/chair/chair_0.1_best_cd.pth'}, 'dataset': {'type': 'shapenet_r2n2', 'eval_split': 'val', 'max_points': 4096, 'image_size': 224, 'scale_factor': 1.0, 'subset_ratio': 0.1, 'restrict_model_ids': None, 'root': '[PROJECT_ROOT]/MESC-3D-B/experiment/data/ShapeNet/ShapeNetCore.v2.PC15k', 'r2n2_dir': '[PROJECT_ROOT]/MESC-3D-B/experiment/data/ShapeNet/ShapeNetCore.R2N2', 'pc_dict': 'pc_dict_v2.json', 'split_file': 'R2N2_split.json', 'views_rel_path': 'ShapeNetRendering', 'which_view_from24': '00', 'category': ['chair'], 'mask_images': '${model.use_mask}', 'start_ratio': 0.0}, 'augmentations': {'random_rotation': False, 'data_augmentation_axis_rotation': False, 'data_augmentation_random_flips': False, 'random_translation': False, 'anisotropic_scaling': False}, 'dataloader': {'batch_size': 24, 'num_workers': 8}, 'optimizer': {'type': 'Adam', 'kwargs': {'betas': [0.9, 0.999], 'eps': 1e-06, 'lr': 0.0001}}, 'scheduler': {'type': 'CosLR', 'kwargs': {'epochs': 250, 'initial_epochs': 10}}, 'trainer': {'name': 'ProTrainer', 'gpus': [0], 'start_epoch': 0, 'epoch': 400, 'epoch_interval_to_save': 5, 'epoch_interval_to_val': 1, 'distributed': False, 'demo': False, 'sample': True}, 'path': {'dir_outpath': '[PROJECT_ROOT]/MESC-3D-B/experiment/dir_log_shapenet/', 'dir_checkpoints': None, 'output_dir': '[PROJECT_ROOT]/MESC-3D-B/experiment/dir_out', 'clip_ckpt_path': '[PROJECT_ROOT]/MESC-3D-B/experiment/checkpoints/ViT-B-16.pt', 'ulip_ckpt_path': '[PROJECT_ROOT]/MESC-3D-B/experiment/checkpoints/checkpoint_pointbert.pt', 'ulip_config_path': '[PROJECT_ROOT]/MESC-3D-B/experiment/config/ULIP.yaml', 'prompt_ckpt_path': '[PROJECT_ROOT]/MESC-3D-B/experiment/ckpt_path/LtpModel/shapenet_r2n2/all/all_0.1_prompt.pth', 'text_encoder_ckpt_path': '[PROJECT_ROOT]/MESC-3D-B/experiment/ckpt_path/LtpModel/shapenet_r2n2/all/all_0.1_text_encoder.pth'), 'root_dir': '[PROJECT_ROOT]/MESC-3D-B/experiment'}
2025-11-10 17:06:46,545 - INFO - Random Seed: 42
2025-11-10 17:06:48,177 - INFO - Test dataset size: 56
2025-11-10 17:06:49,103 - INFO - unexpected_keys
2025-11-10 17:06:49,104 - INFO - The checkpoint state_dict contains keys that are not used by the model:
 □[35mtfts_gamma_1□[0m
 □[35mtfts_beta_1□[0m
 □[35mtfts_gamma_2□[0m
 □[35mtfts_beta_2□[0m
 □[35mblocks.blocks.0.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2, tfts_gamma_3, tfts_beta_3}□[0m
 □[35mblocks.blocks.0.Adapter_MLP.adapter_layer_norm_before.{weight, bias}□[0m
 □[35mblocks.blocks.0.Adapter_MLP.scale.{weight, bias}□[0m
 □[35mblocks.blocks.0.Adapter_MLP.down_proj.{weight, bias}□[0m
 □[35mblocks.blocks.0.Adapter_MLP.up_proj.{weight, bias}□[0m
 □[35mblocks.blocks.0.mlp.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2}□[0m
 □[35mblocks.blocks.0.attn.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2}□[0m
 □[35mblocks.blocks.1.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2, tfts_gamma_3, tfts_beta_3}□[0m
 □[35mblocks.blocks.3.Adapter_MLP.up_proj.{weight, bias}□[0m
 □[35mblocks.blocks.3.mlp.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2}□[0m
 □[35mblocks.blocks.3.attn.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2}□[0m
 □[35mblocks.blocks.4.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2, tfts_gamma_3, tfts_beta_3}□[0m
 □[35mblocks.blocks.4.Adapter_MLP.adapter_layer_norm_before.{weight, bias}□[0m
 □[35mblocks.blocks.4.Adapter_MLP.scale.{weight, bias}□[0m
 □[35mblocks.blocks.4.Adapter_MLP.down_proj.{weight, bias}□[0m
 □[35mblocks.blocks.4.Adapter_MLP.up_proj.{weight, bias}□[0m
 □[35mblocks.blocks.4.mlp.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2}□[0m
 □[35mblocks.blocks.4.attn.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2}□[0m
 □[35mblocks.blocks.5.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2, tfts_gamma_3, tfts_beta_3}□[0m
 □[35mblocks.blocks.5.Adapter_MLP.adapter_layer_norm_before.{weight, bias}□[0m
 □[35mblocks.blocks.5.Adapter_MLP.scale.{weight, bias}□[0m
 □[35mblocks.blocks.5.Adapter_MLP.down_proj.{weight, bias}□[0m
 □[35mblocks.blocks.5.Adapter_MLP.up_proj.{weight, bias}□[0m
 □[35mblocks.blocks.5.mlp.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2}□[0m
 □[35mblocks.blocks.5.attn.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2}□[0m
 □[35mblocks.blocks.6.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2, tfts_gamma_3, tfts_beta_3}□[0m
 □[35mblocks.blocks.6.Adapter_MLP.adapter_layer_norm_before.{weight, bias}□[0m
 □[35mblocks.blocks.6.Adapter_MLP.scale.{weight, bias}□[0m
 □[35mblocks.blocks.6.Adapter_MLP.down_proj.{weight, bias}□[0m
 □[35mblocks.blocks.6.Adapter_MLP.up_proj.{weight, bias}□[0m
 □[35mblocks.blocks.6.mlp.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2}□[0m
 □[35mblocks.blocks.6.attn.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2}□[0m
 □[35mblocks.blocks.7.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2, tfts_gamma_3, tfts_beta_3}□[0m
 □[35mblocks.blocks.7.Adapter_MLP.adapter_layer_norm_before.{weight, bias}□[0m
 □[35mblocks.blocks.7.Adapter_MLP.scale.{weight, bias}□[0m
 □[35mblocks.blocks.7.Adapter_MLP.down_proj.{weight, bias}□[0m
 □[35mblocks.blocks.7.Adapter_MLP.up_proj.{weight, bias}□[0m
 □[35mblocks.blocks.7.mlp.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2}□[0m
 □[35mblocks.blocks.7.attn.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2}□[0m
 □[35mblocks.blocks.8.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2, tfts_gamma_3, tfts_beta_3}□[0m
 □[35mblocks.blocks.8.Adapter_MLP.adapter_layer_norm_before.{weight, bias}□[0m
 □[35mblocks.blocks.11.Adapter_MLP.adapter_layer_norm_before.{weight, bias}□[0m
 □[35mblocks.blocks.11.Adapter_MLP.scale.{weight, bias}□[0m
 □[35mblocks.blocks.11.Adapter_MLP.down_proj.{weight, bias}□[0m
 □[35mblocks.blocks.11.Adapter_MLP.up_proj.{weight, bias}□[0m
 □[35mblocks.blocks.11.mlp.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2}□[0m
 □[35mblocks.blocks.11.attn.{tfts_gamma_1, tfts_beta_1, tfts_gamma_2, tfts_beta_2}□[0m
2025-11-10 17:06:49,104 - INFO - [Transformer] Successful Loading the ckpt from [PROJECT_ROOT]/MESC-3D-B/experiment/checkpoints/modelnet_1k.pth
2025-11-10 17:06:49,606 - INFO - chair_0.1_promodel's previous weights loaded.
2025-11-10 17:06:49,608 - INFO - Model Parameters - Total: 139.19M, Trainable: 74.84M
2025-11-10 17:06:49,609 - INFO - Testing started...
2025-11-10 17:07:08,826 - INFO - ==
2025-11-10 17:07:08,826 - INFO - PERFORMANCE STATISTICS
2025-11-10 17:07:08,826 - INFO - ==
2025-11-10 17:07:08,826 - INFO - Model Parameters - Total: 139.19M, Trainable: 74.84M
2025-11-10 17:07:08,826 - INFO - Total running time: 19.22 seconds
2025-11-10 17:07:08,826 - INFO - Average time per batch: 0.3431 seconds
2025-11-10 17:07:08,826 - INFO - GPU Memory - Peak: 5.01 GB, Current: 549.39 MB
2025-11-10 17:07:08,826 - INFO - GPU Memory - Actual usage: 4.48 GB
2025-11-10 17:07:08,826 - INFO - ==
2025-11-10 17:07:08,826 - INFO - VALIDATION RESULTS
2025-11-10 17:07:08,826 - INFO - ==
2025-11-10 17:07:08,826 - INFO - CD: 0.101505
2025-11-10 17:07:08,826 - INFO - F1 Score: 0.381291
2025-11-10 17:07:08,826 - INFO - ==

Figure 19: The test logs of MESC-3D on 10% ShapeNet-R2N2.

E LIMITATIONS AND BROADER IMPACT

Limitations RGF is evaluated on the established ShapeNet-R2N2 and Pix3D single-view reconstruction protocols, with additional all-category ShapeNet-R2N2 results in Appendix B.1.1. This choice enables controlled comparison with prior point-cloud reconstruction methods under widely used camera, rendering, and evaluation settings. A natural next step is to extend the same evaluation protocol to larger and more diverse asset collections such as Objaverse, where standardized single-view renderings, calibrated cameras, and cleaned reconstruction targets are less readily available. We are also interested in further strengthening the geometric prior used by GI. In the current design, the depth-lifted scaffold is used as a soft conditioning signal rather than a hard reconstruction target, and the threshold analysis in Table 6 suggests stable behavior under moderate prior variation. Future work will explore uncertainty-aware depth fusion and adaptive scaffold selection for more challenging camera layouts, occlusions, and open-world object appearances. Future work may also consider richer evaluation protocols beyond CD and F-Score, such as perceptual or structure-aware metrics for fine-grained geometric details.

Broader Impact Fast single-view 3D reconstruction can benefit robotics, AR/VR, digital content creation, and interactive perception by reducing reconstruction latency and lowering the data requirement for training. Potential negative uses include reconstructing objects or environments without appropriate consent, generating geometry for deceptive digital content, or deploying reconstruction in surveillance-oriented pipelines. The present work is evaluated on object-centric public benchmarks and does not release a high-risk foundation model or scraped human-centered dataset; responsible deployment should still respect data ownership, consent, and downstream safety constraints.

F Comparison with Related Methods

To further clarify RGF's novelty, we supplement comparisons with representative related works (key differences are mentioned in the main text):

F.1 Rectified Flow

RGF builds on the straight-line transport view of Rectified Flow, and we do not claim the generic rectified-flow objective itself as a new contribution. The distinction is that Rectified Flow is a general distribution transport and generation framework, whereas RGF instantiates deterministic transport for an image-conditioned single-view 3D inverse problem. In this setting, the model must align 2D image evidence with 3D geometry, resolve single-view depth ambiguity, and output a reconstruction rather than an unconstrained sample from a target distribution. Therefore, the novelty of RGF lies in the task-specific reconstruction framework that couples the straight-line flow objective with Geometric Injection (GI), Intrinsic Geometry Aggregation (IGA), and a single-step consistency model.

Table 27: Key differences between generic Rectified Flow and RGF. RGF uses the straight-line transport principle as a foundation, but adapts it to single-view point-cloud reconstruction through explicit geometric conditioning and manifold-aware feature aggregation.

Aspect	**Rectified Flow**	**RGF**
Problem setting	Generic distribution transport and generation	Image-conditioned single-view 3D reconstruction
Input condition	Usually unconditional or generic conditioning	Single RGB image with depth-lifted geometric priors
Output representation	General data samples in the target domain	3D point cloud reconstruction aligned with the input image
Core objective	Learn a straight transport field between source and target distributions	Learn geometry-conditioned deterministic transport from noise to the target point cloud
Geometric reasoning	No task-specific 2D-to-3D geometric scaffold	GI provides explicit 3D support and IGA aligns features along local surface geometry
Contribution focus	General flow formulation and fast sampling principle	Task-specific reconstruction system with data-efficient, single-step low-latency inference

F.2 PSF & MFM-Point

- **Task Gap:** Both PSF and MFM-Point focus on **3D point cloud generation** (unconditional or class-conditional generation), aiming to map noise to diverse shapes. In contrast, RGF targets **image-conditioned single-view 3D reconstruction**, where the core challenge is not diversity, but the precise geometric alignment between the 2D input image and the recovered 3D structure.
- **Technical Differences:**
  - **Flow Conditioning:** PSF optimizes a straight transport trajectory for fast unconditional generation, and MFM-Point employs a coarse-to-fine multi-scale strategy for generation stability. RGF, however, introduces **Geometric Injection (GI)**, which explicitly embeds depth-based geometric priors into the flow field. Unlike the noise-driven initialization in PSF/MFM-Point, RGF's flow is guided by a deterministic geometric scaffold to resolve single-view depth ambiguity.
  - **Feature Interaction:** MFM-Point relies on standard Euclidean downsampling/upsampling for scale transitions. RGF incorporates **Tangent-Aligned Feature Interaction**, utilizing parallel transport to aggregate features along the point cloud manifold, thereby preventing the geometric dissonance common in standard transport architectures.

F.3 TRANSPORT-AWARE POINT PROCESSING

Parallel Transport Convolution (PTC) develops deformable convolutional operators for data with manifold structure, while NPTC-net specializes the narrow-band transport construction to point clouds Schonsheck et al. (2022); Jin et al. (2022). These studies motivate comparing features only after accounting for the local frame in which they are represented. RGF follows this transport-aware principle but addresses a different setting: image-conditioned single-view reconstruction with an explicit depth scaffold, a learned base–fiber split, and an endpoint-conditioned $SO(d_f)$ operator integrated into rectified and consistency training. The contribution is therefore the reconstruction framework and its integration of geometric conditioning, rather than the invention of parallel transport or transport-aware convolution.

Recent one-step image-conditioned methods such as Point-MF Baba & Yanai (2026) and mean-flow point generators such as MFM-Point Molodyk et al. (2025) provide complementary context at the level of transport objectives. Their input, point-count, preprocessing, and evaluation settings are not aligned with the present calibrated reconstruction protocol. We therefore describe their objective-level relationship without importing an accuracy or latency number from a different protocol; a direct same-protocol comparison remains outside the evidence reported here.

Fast 3D flow-matching methods such as PSF Wu et al. (2023), MFM-Point Molodyk et al. (2025), and Point-MF Baba & Yanai (2026) share the goal of reducing iterative sampling, but they differ in task conditioning and evaluation protocol. The cited generation methods primarily study point-cloud synthesis or a different image-to-point setting, whereas RGF couples RGB, camera metadata, a depth-derived scaffold, and local frame transport for calibrated single-view reconstruction. This distinction supports a mechanism-level comparison only; published speed or quality values from those protocols are not transferred to the present reconstruction tables.

F.4 PC2, CCD-3DR & BDM

- **Core Mechanism Distinction:** These diffusion methods rely on **Stochastic Differential Equations (SDEs)** (standard Diffusion Models), which necessitate computationally expensive iterative sampling (typically hundreds of steps) and large-scale paired data for convergence. RGF adopts **Geometry-Conditioned Rectified Transport** with ODE trajectories, enabling deterministic, single-step, low-latency inference. In the matched PDM comparison, RGF is 11.4$\times$ faster while using more GPU memory and retaining a quality–efficiency trade-off across metrics; the larger ratios against earlier baselines are reported only as historical context.
- **Technical Design Differences:**
  - **Conditioning Strategy:** PC^2 and CCD-3DR utilize **projection conditioning**, which projects 3D points onto 2D image features. This often leads to "ray ambiguity" where points along the same camera ray map to identical features. RGF resolves this via **Geometric Injection**, which uses a depth-unprojected scaffold to provide explicit 3D spatial cues, enforcing depth consistency.
  - **Optimization Target:** BDM attempts to fuse top-down priors with bottom-up reconstruction via complex coupled diffusion processes. RGF simplifies the learning objective into a direct velocity field regression (Rectified Flow Liu et al. (2023b)), which naturally aligns the noise-to-data trajectory without the need for complex multi-stage fusion or distinct prior models.

F.5 CONSISTENCY-ORIENTED AND FEED-FORWARD RECONSTRUCTION

Consistency models formulate a direct relation between noisy states at different noise levels to reduce the number of denoising evaluations Song et al. (2023). Consistency Diffusion Models for single-image 3D reconstruction incorporate image and 3D priors into a diffusion consistency objective Jiang et al. (2025). RGF shares the goal of reducing iterative inference, but its consistency stage distils an image-conditioned rectified transport field whose state already contains a depth-derived scaffold and transport-aware local features. The distinction is therefore in both the learned target and the conditioned state: the former is a solver-generated endpoint of the rectified field, while the latter is reconstructed through GI and IGA rather than through a diffusion posterior-consistency objective.

DiffPoint uses ViT tokens and CLIP conditioning within a point-cloud diffusion framework Feng et al. (2024). Other fast single-view systems predict alternative 3D representations, including Gaussian or triplane-based outputs Szymanowicz et al. (2024); Zou et al. (2024). These methods address the same latency motivation but do not share the present point-count, depth-scaffold, camera, or point-set evaluation protocol. A reported sub-second representation-specific forward time consequently cannot be interpreted as an end-to-end replacement for the 1.825-second RGF path without matching preprocessing, output conversion, hardware, and evaluation stages.

The comparisons in this appendix therefore distinguish mechanism-level positioning from numerical benchmarking. The deterministic Attribute Flow control and the matched PDM table provide aligned non-diffusion and recent single-view point-cloud references, respectively. No numerical superiority is inferred for an unaligned feed-forward, Gaussian, triplane, or external consistency implementation.

F.6 MESC-3D

- **Focus Distinction:** MESC-3D emphasizes **Semantic Mining**, utilizing CLIP-based text prompts and contrastive learning to hallucinate shape details from semantic cues. RGF prioritizes **Geometric Fidelity**, focusing on recovering the intrinsic manifold structure of the object.
- **Technical Innovations:**
  - **Feature Aggregation:** MESC-3D employs standard Transformers (MIT) to mix global semantic and point features. RGF argues that standard attention neglects the non-Euclidean nature of 3D surfaces and instead uses **Intrinsic Geometry Aggregation (IGA)** to align features locally within the tangent space, preserving fine-grained topological details that semantic-based methods often oversmooth.

F.7 GECO

GECO is related to RGF in pursuing efficient image-conditioned 3D generation, but it is not a direct counterpart to our setting. GECO targets image-to-3D asset generation and relies on a multi-view image intermediate followed by a second-stage 3D distillation process. In contrast, RGF directly addresses single-view 3D reconstruction and predicts a point cloud through a deterministic, geometry-conditioned transport process. Thus, the two methods differ in problem setting, output representation, intermediate representation, and core mechanism.

Table 28: Key differences between GECO and RGF. GECO focuses on fast image-to-3D asset generation through multi-view image synthesis and distillation, whereas RGF directly reconstructs point clouds from a single image with explicit geometric conditioning.

Aspect	**GECO**	**RGF**
Problem setting	Image-to-3D generation	Single-view 3D reconstruction
Output representation	Textured mesh	Point cloud
Intermediate representation	Multi-view images	No multi-view image intermediate
Main pipeline	Two-stage distillation	Direct reconstruction framework
Core mechanism	One-step multi-view image generator plus second-stage 3D distillation	Explicit geometric conditioning with deterministic single-step flow matching

G PRELIMINARY

G.1 CONDITIONAL POINT CLOUD DIFFUSION MODEL

In point cloud generation tasks, diffusion models model data evolution by progressively introducing noise through a Markov chain. As time progresses, the original point cloud data is gradually covered by Gaussian noise. Each pair of consecutive time steps follows a transition rule that defines the relationship between point cloud states. Let the initial point cloud be $X_0 \in \mathbb{R}^{N\times 3}$, sampled from the real data distribution p_{data}, consisting of N 3D points. The core objective of the diffusion process is to gradually convert the original point cloud X_0 into a noisy point cloud X_T through a series of noise addition steps. This process can be described by the following Markov chain:

$$q(X_{1:T} \mid X_0) = \prod_{t=1}^{T} q(X_t \mid X_{t-1}), \tag{23}$$

where each time step t's transition distribution $q(X_t \mid X_{t-1})$ follows a Gaussian distribution with mean $\sqrt{1-\beta_t}X_{t-1}$ and variance $\beta_t I$. The parameter β_t is a predefined hyperparameter, and as t increases, the noise intensity increases. To further express this process, we define X_t as a function of the initial point cloud X_0:

$$q(X_t \mid X_0) = \mathcal{N}(X_t; \bar{\alpha}_t X_0, (1-\bar{\alpha}_t)I), \tag{24}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, and $\alpha_t = 1 - \beta_t$ for each time step. As t increases, $\bar{\alpha}_t$ approaches zero, causing $q(X_t \mid X_0)$ to converge to a pure noise distribution.

The reverse denoising process aims to recover the original clean point cloud X_0 from the final noisy point cloud X_T. The reverse process gradually predicts and removes noise using a neural network, restoring the point cloud structure. This process depends not only on the current point cloud X_t but also on external conditions c, such as images or other related data. The reverse process's transition distribution is represented as:

$$p_\theta(X_{0:T}, c) = p(X_T) \prod_{t=1}^{T} p_\theta(X_{t-1} \mid X_t, c), \tag{25}$$

where $p_\theta(X_{t-1} \mid X_t, c)$ is also a Gaussian distribution, with the mean predicted by the neural network to represent the recovery result conditioned on c. Specifically, the transition distribution is:

$$p_\theta(X_{t-1} \mid X_t, c) = \mathcal{N}(X_{t-1}; \mu_\theta(X_t, t, c), \sigma_t^2 I), \tag{26}$$

where $\mu_\theta(X_t, t, c)$ is the network output, indicating the mean recovered from the noisy point cloud X_t and condition c, and σ_t^2 is the noise variance dependent on the time step, controlling the extent of the denoising process.

During training, the neural network minimizes the difference between predicted and actual noise to gradually improve its denoising ability. Since noise decreases with each time step, the network must adapt its output based on the time step and condition to achieve optimal denoising performance. The key to conditional point cloud generation is how effectively external conditions c guide the generation process. Variational inference maximizes the log-likelihood of the data. During training, the model is optimized using the variational lower bound (VLB), which is expressed as:

$$\begin{aligned} \mathcal{L}_{\text{VLB}} =& \mathbb{E}_q \left[-\log p_\theta(X_0 \mid X_1, c) + D_{\text{KL}}(q(X_T \mid X_0) \| p(X_T)) \right. \\ & \left. + \sum_{t=2}^{T} D_{\text{KL}}(q(X_{t-1} \mid X_t, X_0) \| p_\theta(X_{t-1} \mid X_t, c)) \right] \end{aligned} \tag{27}$$

where $D_{\text{KL}}(\cdot\|\cdot)$ is the Kullback-Leibler divergence, measuring the difference between distributions. By optimizing this training objective, the model finds the best match between the diffusion and reverse denoising processes, achieving high-quality conditional point cloud generation.

To improve training stability and simplify optimization, a simplified mean squared error loss is often used in place of the complex variational lower bound. The loss function is:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,X_0,c,\epsilon}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\sqrt{\bar{\alpha}_t}}\epsilon, c, t)\|^2\right] \tag{28}$$

where $\epsilon \sim \mathcal{N}(0, I)$ is standard Gaussian noise, and $\epsilon_\theta(\cdot)$ is the neural network's output, representing the predicted noise. Optimizing this loss function allows the model to restore clean, conditionally accurate point clouds during the denoising process, resulting in efficient and accurate conditional point cloud generation.

G.2 Quantitative Results

To quantitatively evaluate the reconstruction quality, we employ two standard metrics: Chamfer Distance (CD) and F-Score@1%.

Chamfer Distance assesses the global geometric discrepancy by computing the bidirectional mean of squared nearest-neighbour distances between the reconstructed point set $\mathcal{P}_{rec}$ and the ground truth $\mathcal{P}_{gt}$. We report this value after multiplication by 10^3:

$$\begin{aligned}\mathcal{L}_{CD}(\mathcal{P}_{rec}, \mathcal{P}_{gt}) =& \frac{1}{|\mathcal{P}_{rec}|}\sum_{x\in\mathcal{P}_{rec}}\min_{y\in\mathcal{P}_{gt}}\|x-y\|_2^2 \\ &+ \frac{1}{|\mathcal{P}_{gt}|}\sum_{y\in\mathcal{P}_{gt}}\min_{x\in\mathcal{P}_{rec}}\|x-y\|_2^2\end{aligned} \tag{29}$$

However, given CD's sensitivity to outliers, we complement it with the F-Score. The evaluator applies the threshold $\tau = 0.01$ to squared nearest-neighbour distances and then computes the harmonic mean of precision and recall. This convention is the source of the "F1@1%" label used in the result tables.

H THEORETICAL FOUNDATIONS OF INTRINSIC GEOMETRY AGGREGATION

This section makes precise the geometric claims used by the Intrinsic Geometry Aggregation (IGA) module. The statements concern the discrete operator implemented in RGF and separate properties that are guaranteed by construction from properties that would require an explicitly equivariant connection network. The notation follows Sec. 3.3: d_b and d_f denote the base and fiber dimensions, respectively. The proofs do not depend on particular numerical choices for these dimensions, the number of bundle candidates, the number of heads, or the neighborhood size; those choices define the finite architecture rather than the validity of the algebraic guarantees.

For reproducibility, the reported configuration uses $d_b = 8$ and $d_f = 32$, with four bundle candidates, four transport-aware attention heads, and an eight-neighbor local neighborhood. These values specify the finite model used for the experiments, whereas the $SO(d_f)$ and norm-preservation statements below hold for any compatible dimensions and neighborhood definition.

H.1 BUNDLE NOTATION AND THE DISCRETE OPERATOR

Let $\mathcal{B} \subseteq \mathbb{R}^{d_b}$ be the learned base-coordinate space. A token is represented by a pair $(\mathbf{b}_i, \boldsymbol{\gamma}_i)$ with $\mathbf{b}_i \in \mathcal{B}$ and $\boldsymbol{\gamma}_i \in \mathcal{F}_{\mathbf{b}_i} \simeq \mathbb{R}^{d_f}$. The latter is a local fiber (or tangent-coordinate) representation; two fibers are not compared directly until one of them is transported to the other. For a matrix A, define

$$\operatorname{skew}(A) = \frac{1}{2}(A - A^\top). \tag{30}$$

The transport network in IGA produces the endpoint-dependent generator

$$S_{ij} = \operatorname{skew}(\operatorname{MLP}([\mathbf{b}_i, \mathbf{b}_j])), \qquad \mathbf{P}_{ij} = \exp(S_{ij}), \tag{31}$$

where $[\cdot,\cdot]$ denotes concatenation. In the implementation, transport acts on the fiber before the learned head projections. Let H be the number of heads and $d_h = d_m/H$ the per-head dimension. For head h,

$$\begin{aligned} \mathbf{q}_i^{(h)} &= \mathbf{W}_q^{(h)} \widetilde{\boldsymbol{\gamma}}_i, \\ \mathbf{k}_{ij}^{(h)} &= \mathbf{W}_k^{(h)} \big(\mathbf{P}_{ij} \widetilde{\boldsymbol{\gamma}}_j\big), \\ \mathbf{v}_{ij}^{(h)} &= \mathbf{W}_v^{(h)} \big(\mathbf{P}_{ij} \widetilde{\boldsymbol{\gamma}}_j\big), \end{aligned} \tag{32}$$

where $\mathbf{W}_{q,k,v}^{(h)} : \mathbb{R}^{d_f} \to \mathbb{R}^{d_h}$. For the frame-covariance calculations below, we use the equivalent fiber-space operator notation $\mathbf{q}_i, \mathbf{k}_j, \mathbf{v}_j \in \mathbb{R}^{d_f}$ with an arbitrary positive scale σ; the implementation uses the per-head scale $\sigma = \sqrt{d_h}$ after projection. The attention logit, weight, and output at token i are

$$\begin{aligned} a_{ij} &= \frac{\mathbf{q}_i^\top \mathbf{P}_{ij} \mathbf{k}_j}{\sigma}, \qquad & \alpha_{ij} &= \frac{\exp(a_{ij})}{\sum_{\ell \in \mathcal{N}(i)} \exp(a_{i\ell})}, \\ \mathbf{o}_i &= \sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{P}_{ij} \mathbf{v}_j. \end{aligned} \tag{33}$$

H.2 TRANSPORT AND ATTENTION GUARANTEES

Lemma H.1 (Isometric parallel transport)**.** *For every pair of tokens, the operator in Eq. 31 satisfies* $\mathbf{P}_{ij} \in SO(d_f)$*. Consequently, for any* $\mathbf{u}, \mathbf{w} \in \mathbb{R}^{d_f}$*,*

$$\langle \mathbf{P}_{ij}\mathbf{u}, \mathbf{P}_{ij}\mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{w} \rangle, \qquad \|\mathbf{P}_{ij}\mathbf{u}\|_2 = \|\mathbf{u}\|_2. \tag{34}$$

Proof. By Eq. 30, $S_{ij}^\top = -S_{ij}$. The transpose of a matrix exponential is the exponential of the transpose, and the exponential of a skew-symmetric matrix is orthogonal. Therefore,

$$\mathbf{P}_{ij}^\top \mathbf{P}_{ij} = \exp(S_{ij})^\top \exp(S_{ij}) = \exp(-S_{ij}) \exp(S_{ij}) = \mathbf{I}. \tag{35}$$

Moreover, $\det(\exp(S_{ij})) = \exp(\operatorname{tr}(S_{ij})) = 1$, because a skew-symmetric matrix has zero trace. Hence $\mathbf{P}_{ij} \in SO(d_f)$. Equation 34 follows from $\mathbf{P}_{ij}^\top \mathbf{P}_{ij} = \mathbf{I}$. □

The preceding lemma is an exact property of the implementation. In particular, the transport can rotate or reflect neither the length nor the inner product of a fiber feature; the determinant-one constraint restricts it to orientation-preserving rotations.

In finite-precision execution, the corresponding numerical diagnostic is

$$\varepsilon_{\mathrm{orth}}(i,j) = \left\| \mathbf{P}_{ij}^{\top}\mathbf{P}_{ij} - \mathbf{I} \right\|_F . \tag{36}$$

The skew-symmetric exponential makes this quantity a round-off diagnostic rather than a learned constraint: in exact arithmetic it is zero for every pair. An observed value can identify implementation or precision error, but it cannot by itself establish that the learned base coordinates correspond to a physical tangent bundle.

Theorem H.2 (Invariance under local orthogonal frames)**.** *Suppose a change of fiber frames is represented by orthogonal matrices $\mathbf{R}_i, \mathbf{R}_j \in O(d_f)$. If the local vectors and transport transform covariantly,*

$$\mathbf{q}_i' = \mathbf{R}_i\mathbf{q}_i, \qquad \mathbf{k}_j' = \mathbf{R}_j\mathbf{k}_j, \qquad \mathbf{P}_{ij}' = \mathbf{R}_i\mathbf{P}_{ij}\mathbf{R}_j^{\top}, \tag{37}$$

then the pre-softmax score and the softmax weight are unchanged, and the attention output transforms equivariantly:

$$a_{ij}' = a_{ij}, \qquad \alpha_{ij}' = \alpha_{ij}, \qquad \mathbf{o}_i' = \mathbf{R}_i\mathbf{o}_i. \tag{38}$$

Proof. Substituting Eq. 37 into Eq. 33 gives

$$\begin{aligned} \mathbf{q}_i'^{\top}\mathbf{P}_{ij}'\mathbf{k}_j' &= (\mathbf{R}_i\mathbf{q}_i)^{\top}(\mathbf{R}_i\mathbf{P}_{ij}\mathbf{R}_j^{\top})(\mathbf{R}_j\mathbf{k}_j) \\ &= \mathbf{q}_i^{\top}\mathbf{R}_i^{\top}\mathbf{R}_i\mathbf{P}_{ij}\mathbf{R}_j^{\top}\mathbf{R}_j\mathbf{k}_j = \mathbf{q}_i^{\top}\mathbf{P}_{ij}\mathbf{k}_j. \end{aligned} \tag{39}$$

Thus $a_{ij}' = a_{ij}$ and the softmax weights are identical. If values transform as $\mathbf{v}_j' = \mathbf{R}_j\mathbf{v}_j$, then

$$\mathbf{o}_i' = \sum_j \alpha_{ij}\mathbf{R}_i\mathbf{P}_{ij}\mathbf{R}_j^{\top}\mathbf{R}_j\mathbf{v}_j = \mathbf{R}_i\sum_j \alpha_{ij}\mathbf{P}_{ij}\mathbf{v}_j = \mathbf{R}_i\mathbf{o}_i. \tag{40}$$

□

The theorem states the coordinate-free property targeted by transport-then-attend. It is conditional on the covariance rule in Eq. 37; the current endpoint MLP is not itself constrained to be equivariant to arbitrary changes of fiber frame. What RGF enforces exactly is the isometric parameterization in Lemma H.1. This distinction prevents the implementation from being interpreted as a proof of full gauge equivariance.

Proposition H.3 (Perturbation bound for the attention logit)**.** *Let $\mathbf{P}_{ij} = \mathbf{I} + \mathbf{A}_{ij}$ and let $a_{ij}^{(0)} = \mathbf{q}_i^{\top}\mathbf{k}_j/\sigma$ be the score obtained with identity transport. For fixed $\mathbf{q}_i$ and $\mathbf{k}_j$,*

$$\left| a_{ij} - a_{ij}^{(0)} \right| \le \frac{\|\mathbf{q}_i\|_2 \, \|\mathbf{A}_{ij}\|_2 \, \|\mathbf{k}_j\|_2}{\sigma}. \tag{41}$$

Proof. Direct subtraction gives

$$a_{ij} - a_{ij}^{(0)} = \frac{\mathbf{q}_i^{\top}\mathbf{A}_{ij}\mathbf{k}_j}{\sigma}. \tag{42}$$

The Cauchy–Schwarz inequality and the definition of the operator norm imply

$$\left|\mathbf{q}_i^{\top}\mathbf{A}_{ij}\mathbf{k}_j\right| \le \|\mathbf{q}_i\|_2 \, \|\mathbf{A}_{ij}\mathbf{k}_j\|_2 \le \|\mathbf{q}_i\|_2 \, \|\mathbf{A}_{ij}\|_2 \, \|\mathbf{k}_j\|_2, \tag{43}$$

which proves Eq. 41. □

For the exponential parameterization, $\mathbf{A}_{ij} = \exp(S_{ij}) - \mathbf{I}$ and therefore

$$\|\mathbf{A}_{ij}\|_2 \le \exp(\|S_{ij}\|_2) - 1. \tag{44}$$

When the last layer of the transport MLP is initialized near zero, $\exp(S_{ij}) = \mathbf{I} + S_{ij} + O(\|S_{ij}\|_2^2)$, so Proposition H.3 explains why the initial attention perturbation is small while the exact isometry from Lemma H.1 is retained throughout training.

H.3 CURVATURE CORRECTION AND STABILITY

For completeness, a differential-geometric interpretation can be obtained by considering a differentiable connection field $\{\mathbf{\Gamma}_r(\mathbf{b})\}_{r=1}^{d_b}$ acting on the fiber. Its curvature components would be

$$\mathbf{\Omega}_{rs}(\mathbf{b}) = \frac{\partial \mathbf{\Gamma}_s}{\partial b_r}(\mathbf{b}) - \frac{\partial \mathbf{\Gamma}_r}{\partial b_s}(\mathbf{b}) + [\mathbf{\Gamma}_r(\mathbf{b}), \mathbf{\Gamma}_s(\mathbf{b})], \tag{45}$$

where $[A, B] = AB - BA$. A coordinate-free energy contraction of the curvature is

$$\mathbf{E}_\Omega(\mathbf{b}) = \sum_{1 \le r < s \le d_b} \mathbf{\Omega}_{rs}(\mathbf{b})^\top \mathbf{\Omega}_{rs}(\mathbf{b}). \tag{46}$$

The expression above provides an interpretation rather than an implementation identity. RGF does not numerically recover a Levi–Civita connection, differentiate a connection field, or claim an exact curvature tensor or path-integrated holonomy. The implemented correction is a learned finite-dimensional matrix-valued residual map $\mathbf{R}_\theta(\mathbf{b})$ on the base coordinates:

$$\widetilde{\boldsymbol{\gamma}}(\mathbf{b}) = [\mathbf{I} + \lambda_c \mathbf{R}_\theta(\mathbf{b})]\, \boldsymbol{\gamma}(\mathbf{b}), \tag{47}$$

where λ_c is the residual scale used by the finite model. This construction supplies a bounded-capacity residual only when an explicit norm bound on $\mathbf{R}_\theta$ is assumed; the $SO(d_f)$ guarantee belongs to $\mathbf{P}_{ij}$, not to this curvature-inspired branch.

For the reported finite architecture, $\mathbf{R}_\theta$ is produced by a two-layer MLP $\mathbb{R}^{d_b} \to \mathbb{R}^{64} \to \mathbb{R}^{d_f \times d_f}$ with a GELU intermediate nonlinearity and a zero-initialized final projection. The output is reshaped into the residual matrix in Eq. 47. The connection-field symbols $\mathbf{\Gamma}_r$ and $\mathbf{\Omega}_{rs}$ above are therefore a differential-geometric interpretation, not additional train-time tensors used to claim a physical connection. No torsion-free, Levi–Civita, or full gauge-equivariance constraint is imposed on this residual branch. The exact structural constraint is applied to the separate endpoint transport operator $\mathbf{P}_{ij}$ through skew-symmetrization and the matrix exponential.

The endpoint transport uses a local neighborhood rather than a dense all-token graph. With $d_f = 32$ and the reported eight-neighbor configuration, the matrix exponential is evaluated on local fiber operators generated from base-coordinate pairs. Its exact spectral norm is one in exact arithmetic, while finite-precision error is diagnosed by Eq. 36. This establishes an operator-level norm guarantee; it does not provide a universal complexity bound for arbitrary feature dimensions or neighborhood sizes. The current comparison also does not include an unconstrained-attention or projection-based orthogonalization ablation, so the statement concerns the guarantee supplied by the parameterization rather than an empirical claim that it is optimal for every architecture.

Finite-precision and local-graph boundary. In the reported configuration, the local transport calls are batched 32×32 matrix-exponential evaluations for the eight neighbors of each token; a dense all-token transport graph is not used. The $SO(d_f)$ result is an exact-arithmetic property of the skew-symmetric parameterization. Under finite precision, the relevant quantity is the diagnostic in Eq. 36; no empirical distribution of this error or of composed-transport drift is asserted without an instrumentation run. Thus the construction supplies an algebraic isometry guarantee for the implemented operator, not a measured claim about numerical drift at every hardware precision or tokenization.

The pairwise transport MLP maps the concatenated base coordinates in $\mathbb{R}^{2d_b}$ to a flattened $d_f \times d_f$ generator, which is reshaped, skew-symmetrized, and exponentiated. No separate torsion, metric-compatibility, spectral-norm, or orthogonalization penalty is required for the $SO(d_f)$ property; the only structural operation needed for that property is the skew-symmetrization before the matrix exponential. This parameterization should therefore be understood as an algebraic isometry guarantee, not as an empirical claim that it dominates all unconstrained alternatives.

Lemma H.4 (Conditional residual bound)**.** *If $\|\mathbf{R}_\theta(\mathbf{b})\|_2 \le M_R$ on the relevant base-coordinate region, then the update in Eq. 47 satisfies*

$$\|\widetilde{\boldsymbol{\gamma}}\|_2 \le (1 + |\lambda_c| M_R) \|\boldsymbol{\gamma}\|_2, \qquad \|\widetilde{\boldsymbol{\gamma}} - \boldsymbol{\gamma}\|_2 \le |\lambda_c| M_R \|\boldsymbol{\gamma}\|_2. \tag{48}$$

Proof. The second inequality follows directly from submultiplicativity and the assumed bound on $\mathbf{R}_\theta$. The first follows from the triangle inequality. □

Proposition H.5 (Stability of the learned transport)**.** *Assume that the skew-symmetric generator $S(\mathbf{b}_i, \mathbf{b}_j)$ is L_S-Lipschitz in the concatenated pair of base coordinates. For two pairs $\mathbf{b}$ and $\mathbf{b}'$,*

$$\|\exp(S(\mathbf{b}')) - \exp(S(\mathbf{b}))\|_2 \leq L_S \|\mathbf{b}' - \mathbf{b}\|_2. \tag{49}$$

Proof. For skew-symmetric A and B, the standard integral identity for the matrix exponential gives

$$\exp(A) - \exp(B) = \int_0^1 \exp((1-s)A)(A-B)\exp(sB)\, ds. \tag{50}$$

Both exponential factors are orthogonal, so their operator norms equal one. Taking norms and integrating yields $\|\exp(A) - \exp(B)\|_2 \leq \|A - B\|_2$. Setting $A = S(\mathbf{b}')$ and $B = S(\mathbf{b})$, then applying the Lipschitz assumption, proves Eq. 49. □

H.4 Conditional transport and the scope of the guarantees

Learned coordinate-domain scope. The base coordinates $\mathbf{b}_i$ are learned network outputs, and the fiber features $\boldsymbol{\gamma}_i$ are learned representations. They are not measured surface coordinates, estimated tangent vectors, or a data-derived metric. Accordingly, "manifold", "base", and "curvature-inspired" describe the organization of the feature space used by IGA; they do not establish an isometry between that space and the physical object surface. A physical parallel-transport interpretation would require geometry-derived frames, a specified metric, and path-based validation beyond the current construction.

The geometric results above concern the feature operator. The same conditional-flow notation also gives a simple exact statement for the transport objective. For a paired source and target $(\mathcal{X}_0, \mathcal{X}_1)$, the path in Eq. 2 obeys

$$\frac{d}{dt}\mathcal{X}_t = \mathcal{X}_1 - \mathcal{X}_0. \tag{51}$$

Thus, if the learned field equals this target velocity along the path, uniqueness of the ODE solution implies $\phi_t(\mathcal{X}_0) = \mathcal{X}_t$ and $\phi_1(\mathcal{X}_0) = \mathcal{X}_1$. With finite data and a restricted model class, Eq. 4 instead learns the conditional regression function $\mathbb{E}[\mathcal{X}_1 - \mathcal{X}_0 \mid \mathcal{X}_t, t]$; the equality is exact only in the realizable/no-zero-training-error limit.

The linear path is a design choice for reducing the numerical burden of conditional transport, not a universal necessity theorem for single-view reconstruction. The exact constant-velocity statement applies to the prescribed conditional interpolation, whereas the learned field only approximates its regression target. Likewise, a reduction in Euler discretization error follows from a straighter field only under the usual regularity and solver assumptions; it is not guaranteed for arbitrary conditioning shifts, depth corruption, or camera errors. The NFE table in Sec. C.4 records the empirical operating range of this approximation.

The operator in Eq. 31 is an endpoint-dependent surrogate for parallel transport. A genuine path-dependent connection would additionally satisfy the composition law $\mathbf{P}_{ik}\mathbf{P}_{kj} = \mathbf{P}_{ij}$ for a specified path through j, and nonzero curvature would make transport around a loop path-dependent. RGF does not impose this law or numerically integrate a path; it guarantees the $SO(d_f)$ constraint, the bounds above, and the stated conditional invariance property. A useful diagnostic for future implementations is the triplet discrepancy

$$\Delta_{ijk} = \|\mathbf{P}_{ik}\mathbf{P}_{kj} - \mathbf{P}_{ij}\|_F, \tag{52}$$

which measures, rather than assumes, how closely the learned surrogate follows a composition-preserving connection.

Table 29: Correspondence between the geometric construction and the guarantees established in this appendix.

Geometric concept	**RGF realization**	**Guaranteed property**
Bundle token	$(\mathbf{b}_i, \boldsymbol{\gamma}_i)$	Base/fiber separation
Metric-compatible transport	$\mathbf{P}_{ij} = \exp(\mathrm{skew}(\mathrm{MLP}))$	$\mathbf{P}_{ij} \in SO(d_f)$ and norm preservation
Transported attention	$\mathbf{P}_{ij}\mathbf{k}_j, \mathbf{P}_{ij}\mathbf{v}_j$	Frame-invariant score under the covariance assumptions
Curvature-inspired correction	$(\mathbf{I} + \lambda_c \mathbf{R}_\theta(\mathbf{b}))\boldsymbol{\gamma}$	Conditional residual bound under a norm assumption
Connection surrogate	Endpoint MLP on $(\mathbf{b}_i, \mathbf{b}_j)$	Stable endpoint map; composition is not imposed

I VISUALIZATION OF GEOMETRIC PRIORS

As shown in Figure 20, Figure 21, Figure 22, Figure 23, Figure 24 and Figure 25, we present qualitative results about the generation process of geometric priors that further highlight the superiority of our RGF with geometric injection (GI).

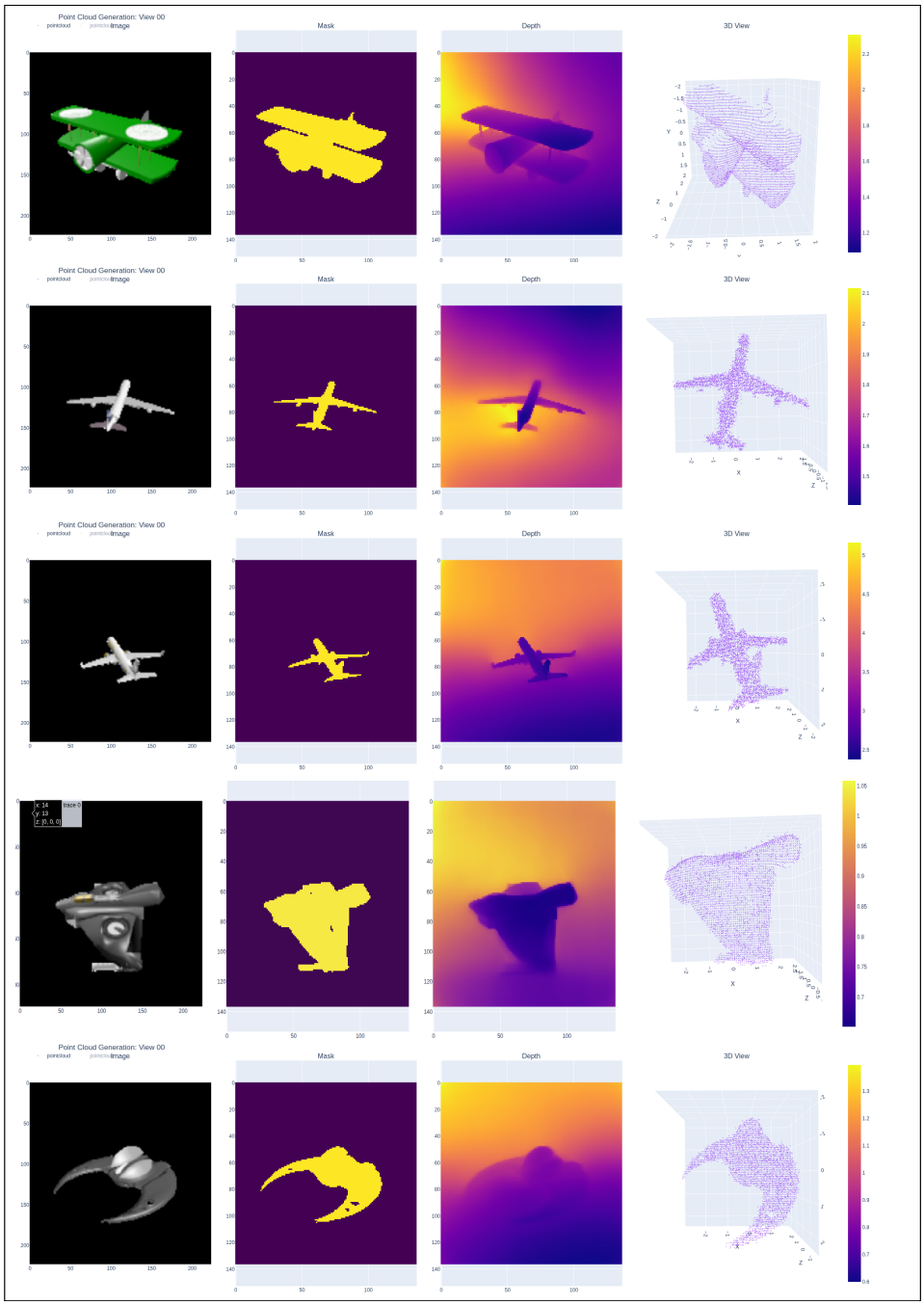


Figure 20: **Qualitative Visualization of Geometric Priors.** From left to right: the input single-view image, the corresponding object mask, the estimated depth map, and the extracted geometric priors.

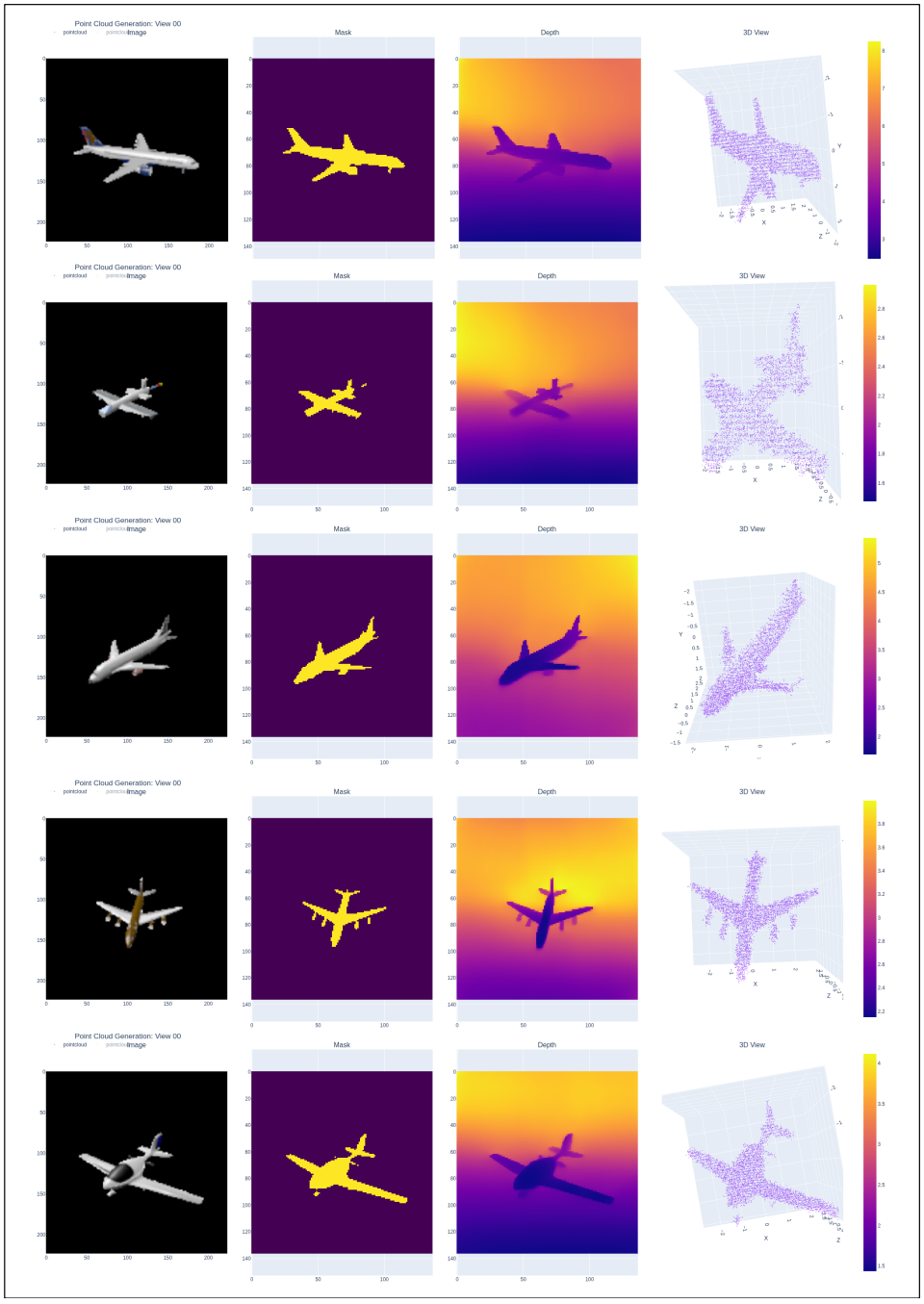


Figure 21: **Qualitative Visualization of Geometric Priors.** From left to right: the input single-view image, the corresponding object mask, the estimated depth map, and the extracted geometric priors.

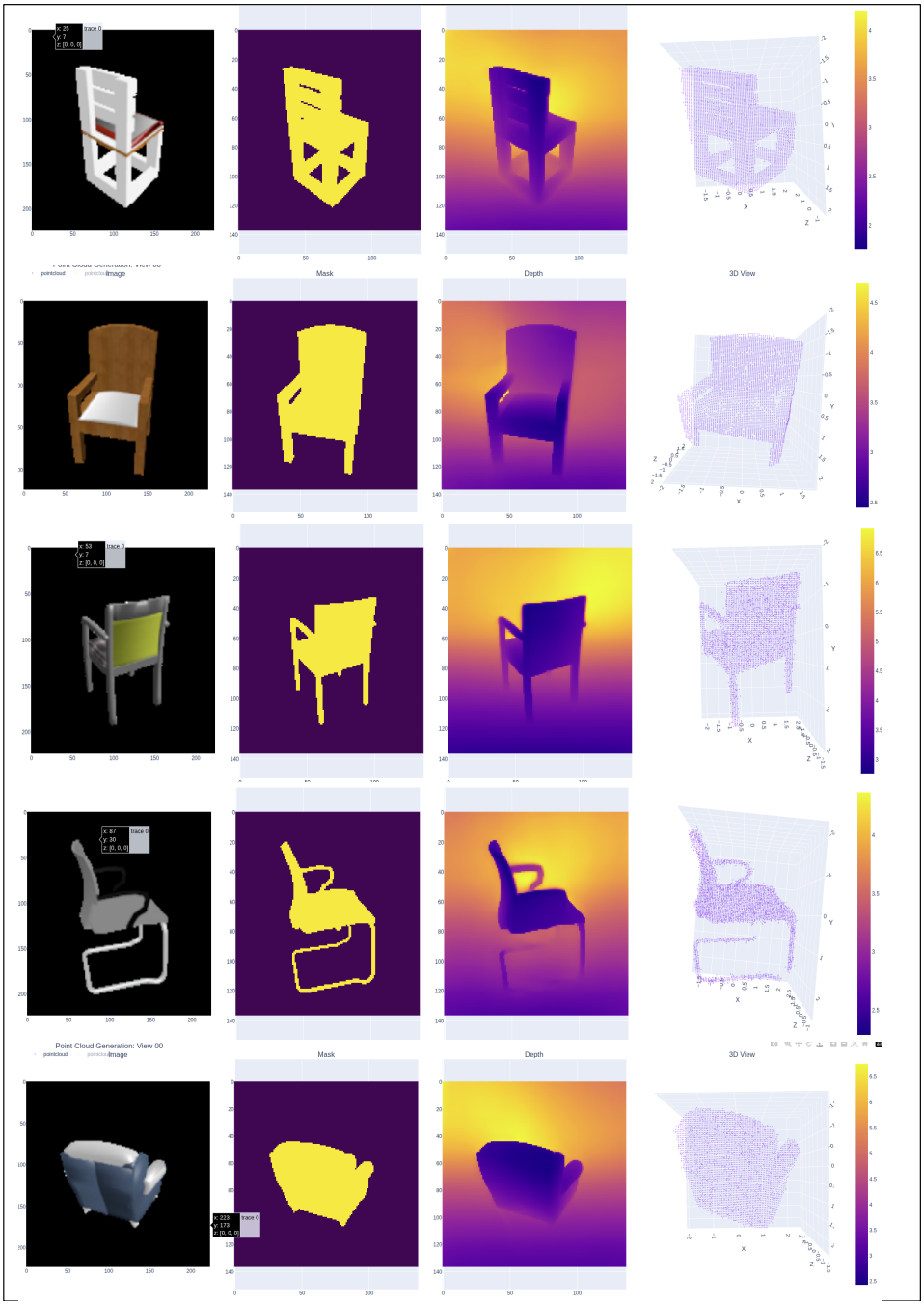


Figure 22: **Qualitative Visualization of Geometric Priors.** From left to right: the input single-view image, the corresponding object mask, the estimated depth map, and the extracted geometric priors.

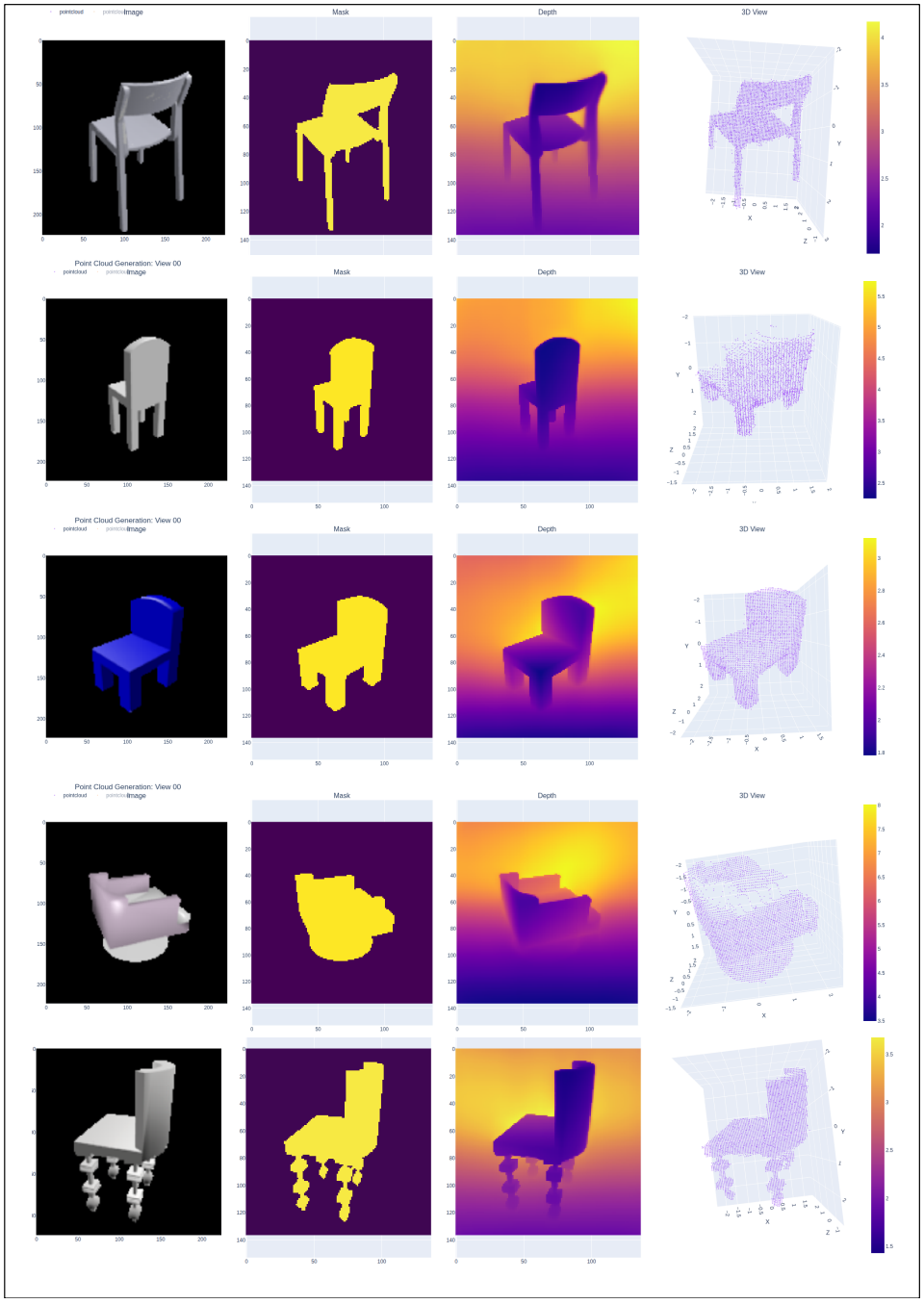


Figure 23: **Qualitative Visualization of Geometric Priors.** From left to right: the input single-view image, the corresponding object mask, the estimated depth map, and the extracted geometric priors.

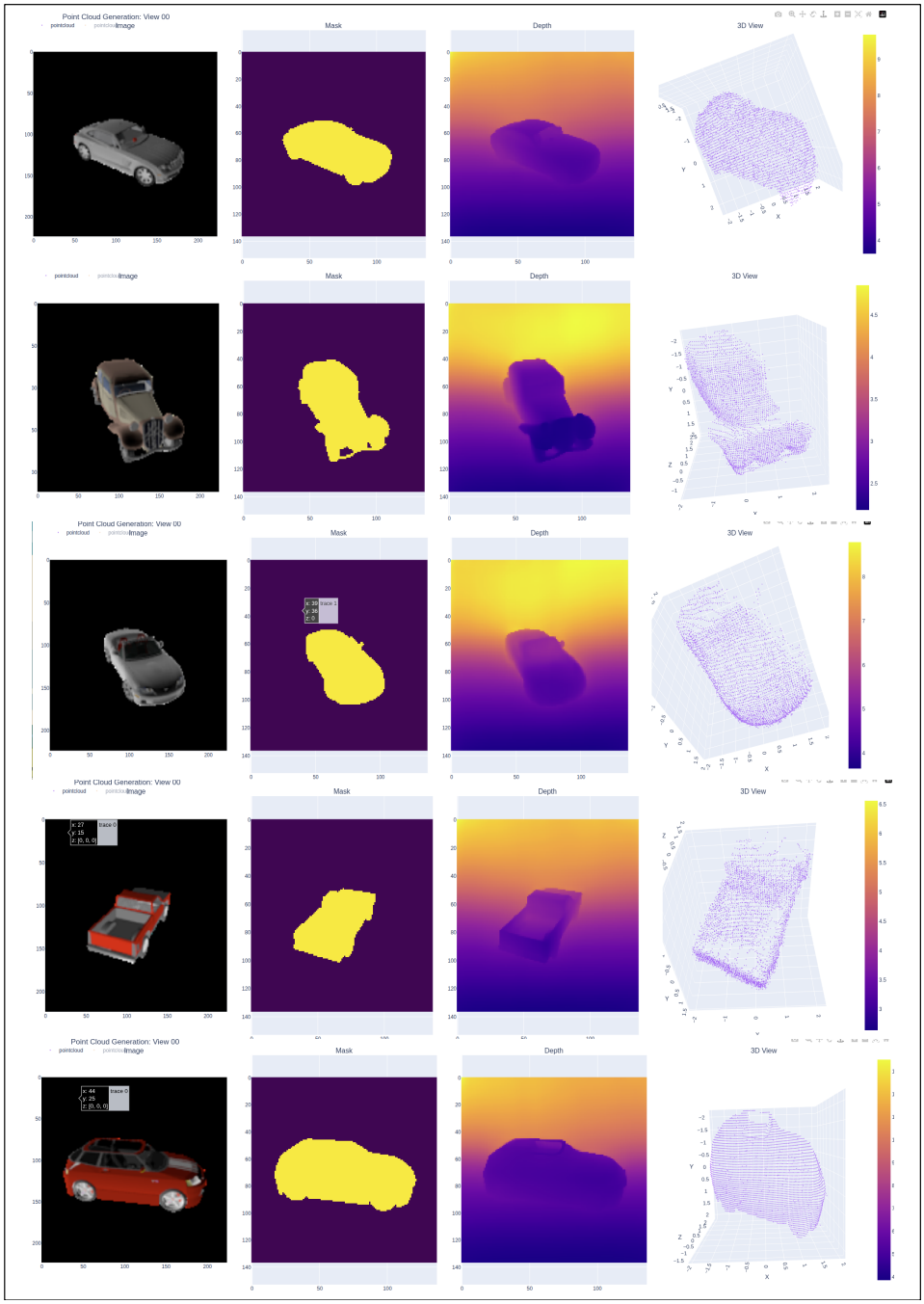


Figure 24: **Qualitative Visualization of Geometric Priors.** From left to right: the input single-view image, the corresponding object mask, the estimated depth map, and the extracted geometric priors.

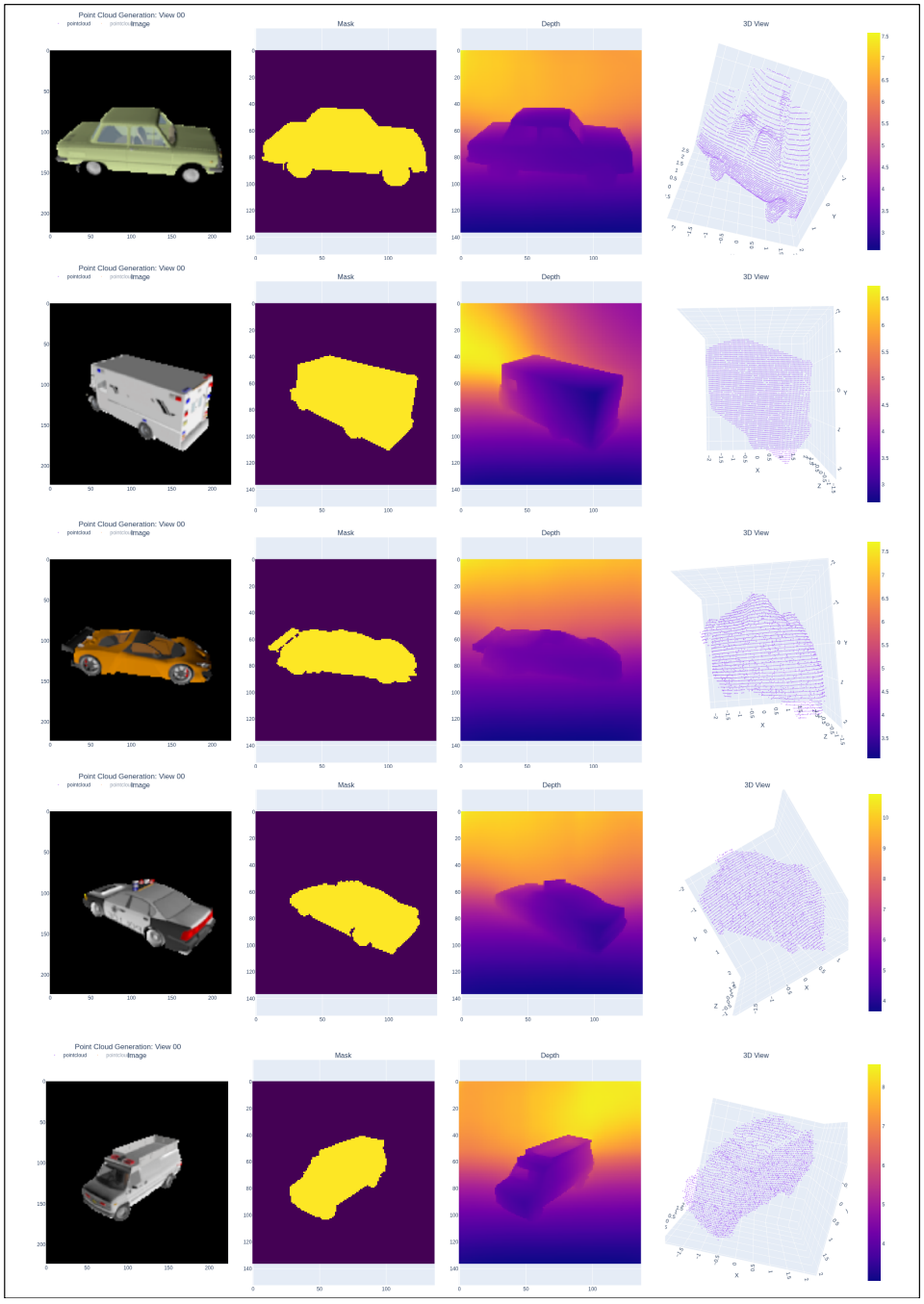


Figure 25: **Qualitative Visualization of Geometric Priors.** From left to right: the input single-view image, the corresponding object mask, the estimated depth map, and the extracted geometric priors.