PRISM: POINT CLOUD RESTORATION FOR IMPAIRED SINGLE-PHOTON DATA VIA LATENT POINT FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-photon imaging enables 3D sensing under low illumination and at long range, making point-cloud restoration essential for converting sparse, contaminated measurements into usable geometry. However, existing object-level datasets offer limited shape diversity and paired reference geometry for evaluating restoration under controlled acquisition conditions. Spatially non-uniform sampling, missing surface observations, and spurious detections further complicate recovery by making geometric evidence unevenly reliable. To address these challenges, we introduce SPORE, a physics-grounded simulated benchmark spanning commonly used ShapeNet categories and three corruption regimes, constructed using ray-based acquisition and Poisson photon statistics. We also propose PRISM, a hierarchical latent flow-matching framework that selectively repairs unreliable local anchors and retains their reliability and correction information to guide subsequent global–local restoration. On SPORE, PRISM achieves the best results on three of four metrics under Mid corruption and all four under High corruption, reducing HD95 by 27.6% and improving F@0.01 by 22.9% relative to the strongest competing result for each metric in the High regime. These findings support reliability-informed restoration as an effective approach to recovering precise geometry under severe simulated single-photon degradation.

1 INTRODUCTION

Single-photon imaging enables 3D sensing under limited illumination and acquisition budgets (Lee et al., 2023), supporting ultralight point-cloud acquisition systems (Ohno et al., 2025) and high-resolution, long-range depth imaging (McCarthy et al., 2025). Yet sparse signal returns and background contamination can produce inaccurate or incomplete surface observations. Reliable point-cloud restoration is therefore essential for translating photon-efficient acquisition into usable geometry for downstream 3D perception.

Object-level restoration requires recovering precise shapes from observations with uneven geometric support. Existing single-photon studies address multi-view surface reconstruction (Luo et al., 2025), scene modeling (Halimi et al., 2026), and uncertainty-aware detection (Goyal et al., 2025), while object-centric acquisition datasets cover relatively few objects and provide reference geometry for only some real captures (Mu et al., 2024). These resources offer limited support for evaluating restoration across diverse object categories with paired clean geometry and controlled sensor-driven corruption. This gap motivates an object-level benchmark that systematically varies degradation severity to assess both global shape preservation and local geometric recovery.

Learning-based point-cloud denoising has advanced through discriminative, score-based, and generative formulations (Rakotosaona et al., 2020; Luo & Hu, 2021b; Mao et al., 2022; Vogel et al., 2024), alongside iterative refinement, adaptive restoration, latent modeling, and structural priors (de Silva Edirimuni et al., 2023; 2024; Mao et al., 2024; Guo et al., 2025; Du et al., 2026; Guo et al., 2026). Much of this progress has been evaluated on synthetic perturbations of sampled surface points. Single-photon degradation introduces a broader challenge by jointly compromising coordinate accuracy, surface coverage, and measurement reliability. When local support is missing

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

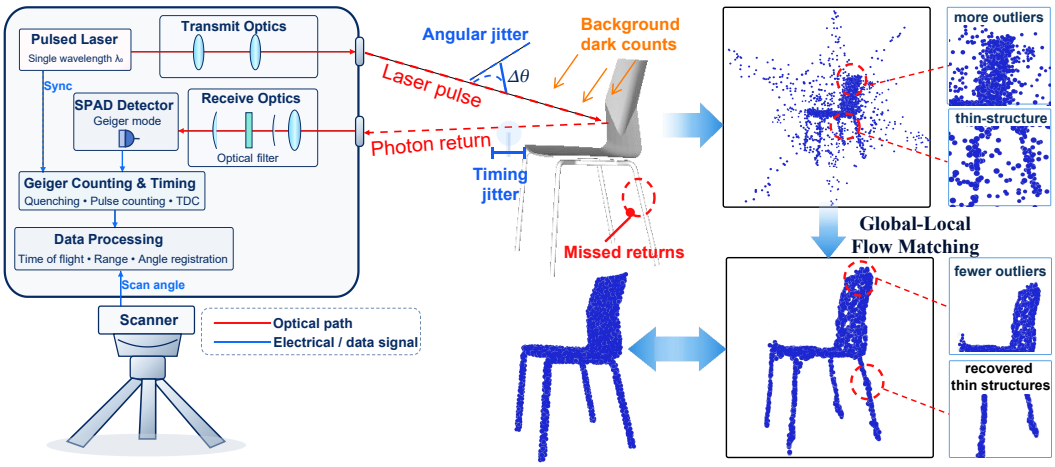


Figure 1: **Single-photon point-cloud degradation and PRISM restoration.** Single-photon acquisition introduces missed returns, spurious detections, and spatial perturbations through background counts, timing uncertainty, and angular jitter. PRISM restores coherent geometry while suppressing severe corruption and recovering thin structures.

or erroneous, restoration must assess which observations remain trustworthy and use that information throughout refinement to guide spatially selective, globally consistent corrections.

We introduce **PRISM** (**P**oint cloud **R**estoration for **I**mpaired **S**ingle-photon data via flow **M**atching), to restore point cloud under severe structured corruption. Operating in a hierarchical latent space (Zeng et al., 2022), PRISM first performs geometry-aware pre-repair, selectively correcting local anchors using reliable neighboring support. It records estimated reliability and applied corrections in explicit token-wise repair states. Repair-state-conditioned global–local flow matching then restores the hierarchical latents while retaining this information throughout transport. Shared repair context conditions both velocity fields, while the evolving global state and token-wise repair information guide local restoration. Our main contributions are:

- **A Physics-Grounded Single-Photon Benchmark.** We introduce SPORE (**S**ingle-**P**hoton **O**bject **R**estoration and **E**valuation), pairing clean geometry with corrupted point clouds generated physics-based acquisition modeling. SPORE enables controlled evaluation across corruption severities.
- **Geometry-Aware Latent Repair.** We introduce reliability-guided anchor correction that produces repaired latent initializations and explicit token-wise repair states, improving robustness to severe corruption.
- **Global-Local Flow Matching.** We preserve repair information through repair-state conditioning and hierarchical latent transport, enhancing the recovery of complex structures.
- **Robust Restoration across High Corruption Regimes.** Evaluation across our benchmark dataset shows competitive performance under mild degradation and increasingly pronounced gains under stronger corruption. Ablations and mechanistic analyses validates the effectiveness of our designed modules.

2 Related Work

Point Cloud Denoising. Single-photon reconstruction has progressed from photon-efficient statistical estimation to methods incorporating learned geometric priors (Kirmani et al., 2014; Shin et al., 2015; Tachella et al., 2019; Halimi et al., 2026). More broadly, point-cloud denoising has evolved from statistical filtering (Rusu & Cousins, 2011) to learned point correction (Rakotosaona et al., 2020; Zhang et al., 2021), joint point-normal estimation (Liu et al., 2024), and iterative refinement (de Silva Edirimuni et al., 2023; 2024). Recent methods exploit multi-scale and long-range structure (Liu et al., 2025; Zhou et al., 2025), adaptive stopping (Guo et al., 2025), dynamic routing (Cheng et al., 2026), and learned structural priors (Guo et al., 2026). These advances motivate restoration that accounts for spatially varying degradation. PRISM explicitly estimates token-wise

geometric reliability, uses reliable neighboring support for latent pre-repair, and retains the resulting repair information throughout subsequent restoration.

Generative Point-cloud Restoration. Generative approaches restore point clouds through learned distributional structure. Representative methods include score-based denoising (Luo & Hu, 2021b), normalizing-flow-based restoration (Mao et al., 2022), diffusion bridges (Vogel et al., 2024), and deterministic diffusion (Liu et al., 2026a). Flow matching learns time-dependent velocity fields for continuous distribution transport (Lipman et al., 2023) and has been applied to point-cloud generation (Hui et al., 2025) and upsampling (Liu et al., 2026b). PRISM formulates restoration as transport from pre-repaired hierarchical latents, conditioning both global and local velocity fields on repair context while using token-wise repair states to guide local interactions and feature modulation.

Latent Representations for Point Clouds. PointFlow models shape and point distributions using continuous normalizing flows (Yang et al., 2019), while latent-conditioned diffusion supports point-cloud generation (Luo & Hu, 2021a). LION introduces a hierarchical representation comprising an object-level global latent and spatially structured local latents (Zeng et al., 2022). For restoration, PD-LTS combines latent modeling with invertible transformations (Mao et al., 2024), and LaPDA adapts latent noise distributions for denoising (Du et al., 2026). PRISM adopts LION's representation to couple global shape inference with local geometric correction. Its contribution lies in reliability-guided latent pre-repair and persistent repair-state conditioning within this representation.

3 Physics-Grounded Single-Photon Point-Cloud Benchmark

We introduce **SPORE**, a simulated benchmark built from ShapeNet meshes (Chang et al., 2015). pairing clean object geometry with single-photon point clouds across controlled corruption levels.

As shown in Figure 1 our simulator combines a pulsed laser, a SPAD detector and a scanner to collect the single-photon point cloud of each normalized mesh. For each normalized mesh, rays are cast from multiple viewpoints of LiDAR, with the nearest valid ray–mesh intersection defining a potential surface return. Signal and background photon counts follow

$$N_s \sim \mathrm{Poisson}(\lambda_s), \qquad N_b \sim \mathrm{Poisson}(\lambda_b), \tag{1}$$

where λ_s depends on photon emission, reflectivity, acquisition geometry, and detection efficiency, while λ_b accounts for ambient background and detector dark counts. Detected surface returns undergo incidence-dependent dropout; background and spurious detections introduce unrelated ranges. Timing, range, and angular uncertainties perturb measurements before conversion to 3D coordinates. Three regimes, Low, Mid, and High jointly vary these factors to increase degradation severity.

SPORE covers 55 categories, with 41,880 training, 5,235 validation, and 5,235 test objects. Each normalized observation and clean target contains 2,048 points, with valid observations across regimes paired to the same clean geometry. An audit of 500 models confirms increasing degradation before and after point-count normalization. Under per-object matching of the one-way mean nearest-neighbor distance to a fixed sampled clean reference, local-density variance is approximately $68.9\times$ and $14.7\times$ that of isotropic Gaussian perturbations under Mid and High corruption, respectively, demonstrating substantially stronger spatial heterogeneity despite similar matched average distance. Simulator settings and additional analyses appear in the Appendix.

4 Method

4.1 Overview

Given an impaired point cloud $\boldsymbol{X}_{\mathrm{n}} \in \mathbb{R}^{N\times 3}$, PRISM reconstructs $\boldsymbol{X}_{\mathrm{p}}$ to approximate the underlying clean geometry $\boldsymbol{X}_{\mathrm{c}}$. As illustrated in Figure 2, an encoder E maps the input into an object-level global latent $\boldsymbol{Z}_g^{\mathrm{n}}$ and a set of spatially organized local latents $\boldsymbol{Z}_\ell^{\mathrm{n}}$ (Zeng et al., 2022). *Geometry-Aware Latent Repair* estimates the reliability of each local latent and corrects its spatial anchor using reliable neighboring observations. It returns repaired latents $\widetilde{\boldsymbol{Z}}_\ell^{\mathrm{n}}$ and states $\mathcal{C}$ that record

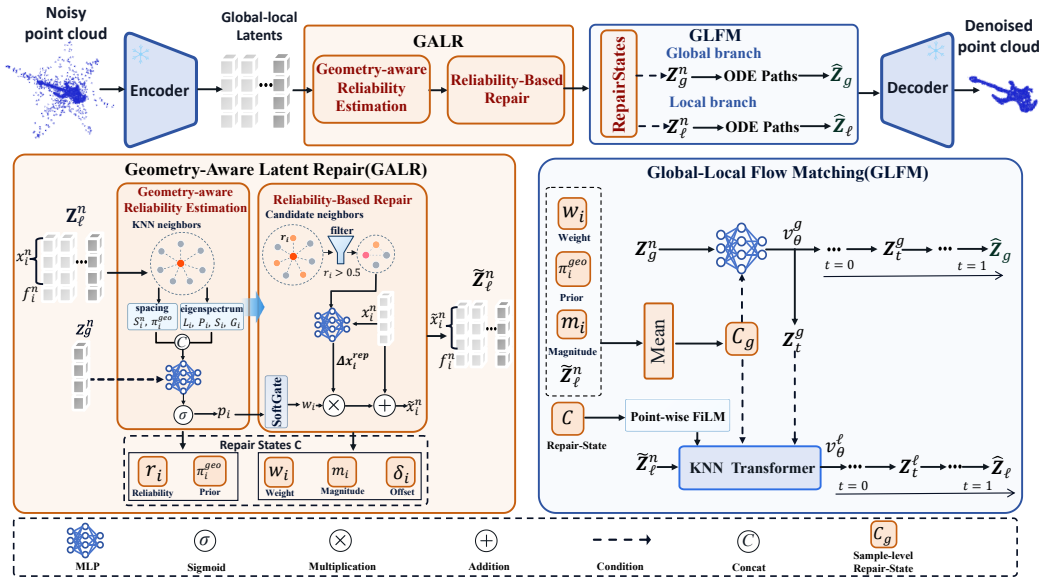


Figure 2: **Overview of PRISM.** Geometry-Aware Latent Repair (GALR) estimates local reliability, repairs unreliable latent anchors, and constructs repair states. Global-Local Flow Matching (GLFM) then restores the hierarchical latent representation under repair-state conditioning before joint decoding.

reliability and applied corrections. *Global-Local Flow Matching* then restores both global and local representation, with the global trajectory and repair states conditioning the local velocity field. The hierachical Flow yields restored latents $(\widehat{\boldsymbol{Z}}_g, \widehat{\boldsymbol{Z}}_\ell)$, which are jointly decoded into the output point cloud $\boldsymbol{X}_{\mathrm{p}} = D(\widehat{\boldsymbol{Z}}_g, \widehat{\boldsymbol{Z}}_\ell)$.

4.2 Geometry-Aware Latent Repair

Impaired single-photon observations produce point clouds with uneven geometric reliability due to photons from background light or backscattering. Geometry-Aware Latent Repair (GALR) aims to estimate this reliability and adaptively repairs local anchors before latent transport. The local representation Z_ℓ^{n} contains M tokens, $\boldsymbol{z}_{\ell,i}^{\mathrm{n}} = [\boldsymbol{x}_i^{\mathrm{n}}, \boldsymbol{f}_i^{\mathrm{n}}]$, where $\boldsymbol{x}_i^{\mathrm{n}} \in \mathbb{R}^3$ is a spatial anchor and $\boldsymbol{f}_i^{\mathrm{n}}$ its feature. GALR updates the anchors while leaving the features unchanged, yielding correction $\widetilde{z}_{\ell,i}^{\mathrm{n}} = [\widetilde{x}_i^{\mathrm{n}}, f_i^{\mathrm{n}}]$ and repair states. The correction assigned to each anchor depends on its estimated reliability.

Geometry-Aware Reliability Estimation. We first measure local spacing as the mean Euclidean distance from each anchor to its K nearest neighbors. Standardizing these distances within the observation gives s_i^{n}, from which we compute a geometric corruptio prior, $\pi_i^{\mathrm{geo}} = \mathrm{sigmoid}((s_i^{\mathrm{n}} - \alpha)/\tau)$. Here, α is a deviation threshold and τ determines transition softness.

Spacing alone cannot distinguish corrupted observations. We therefore supplement this prior with covariance-based linearity L_i, planarity P_i, and scattering S_i descriptors (Weinmann et al., 2015), together with structural coherence $G_i = \max(L_i, P_i)(1 - S_i)$. These quantities form the local geometry descriptor $\boldsymbol{g}_i^{\mathrm{n}} = [s_i^{\mathrm{n}}, \pi_i^{\mathrm{geo}}, L_i, P_i, S_i, G_i]$.

The geometry-derived prior is further calibrated using latent context and object-level corruption statistics:

$$p_i = \mathrm{sigmoid}\left[\mathrm{logit}(\pi_i^{\mathrm{geo}}) + f_\phi\left(\boldsymbol{Z}_{\ell,i}^{\mathrm{n}}, \boldsymbol{Z}_g^{\mathrm{n}}, \boldsymbol{g}_i^{\mathrm{n}}, \boldsymbol{\nu}\right)\right], \qquad r_i = 1 - p_i. \tag{2}$$

The corruption score $p_i \in [0, 1]$ determins reliability r_i. The descriptor $\boldsymbol{\nu}$ contains the mean, standard deviation, top-10% mean, and maximum of the geometric priors within the observation. The estimation combines local geometry, local latent information, global context, and object-level corruption statistics.

Reliability-Based Repair. For each anchor, we construct a repair neighborhood $\mathcal{R}_i$ by retaining up to K_r reliable nearest candidates with corruption probability below a threshold, $p_j < \tau_r$. A relation network predicts normalized weights ω_{ij} from pairwise geometric, latent, and reliability cues. Their weighted coordinates give the estimated geometry-supported anchor $x_i^{\mathrm{geo}} = \sum_{j\in\mathcal{R}_i} \omega_{ij} x_j^{\mathrm{n}}$. The repaired spatial achor is updated by

$$\widetilde{x}_i^{\mathrm{n}} = x_i^{\mathrm{n}} + w_i(x_i^{\mathrm{geo}} - x_i^{\mathrm{n}} + \Delta x_i^{\mathrm{res}}), \qquad w_i = \mathrm{clip}\left(\frac{p_i - \tau_{\mathrm{low}}}{\tau_{\mathrm{high}} - \tau_{\mathrm{low}}}, 0, 1\right), \tag{3}$$

where $\Delta \boldsymbol{x}_i^{\mathrm{res}}$ is predicted from the local latent and its aggregated neighborhood information. The thresholds τ_{low} and τ_{high} determine the transition from no correction to the full proposed update.

The repaired token does not indicate how much it was changed. We retain this information in a repair state as

$$c_i = [r_i,\, w_i,\, \pi_i^{\mathrm{geo}},\, m_i,\, \tanh(\delta_i)]\,, \qquad \delta_i = \widetilde{x}_i^{\mathrm{n}} - x_i^{\mathrm{n}}, \qquad m_i = \tanh\left(\|\delta_i\|_2\right). \tag{4}$$

The state contains the reliability score r_i,repair weight w_i, geometric prior π_i^{geo}, repaired displacement m_i and its corresponding direction $\tanh(\delta_i)$. GALR returns the repaired local latents $\widetilde{Z}_\ell^{\mathrm{n}} = \{\widetilde{z}_{\ell,i}^{\mathrm{n}}\}_{i=1}^M$ together with the repair-state set $C = \{c_i\}_{i=1}^M$ for subsequent Global-Local Flow Matching.

4.3 Global-Local Flow Matching

GALR repairs local spatial anchors while leaving the global latent and local features unchanged. Global-Local Flow Matching (GLFM) restores the hierarchical representation by learning coupled velocity fields from the impaired global latent and repaired local latent their clean counterparts. A shared repair state conditions both fields, while the evolving global latent guides local geometry and latent feature restoration, particularly in regions with limited geometric evidence.

Repair-State Conditioning. We summarize the outcome of GALR into an object-level repair context $\mathbf{c}_g$ by concatenating the mean repaired local token with three averaged repair statistics: the geometric prior, repair weight, and bounded repair magnitude. Specifically,

$$\mathbf{c}_g = \left[\mathrm{Mean}_i\left(\widetilde{\mathbf{z}}_{\ell,i}^{\mathrm{n}}\right), \mathrm{Mean}_i\left(\pi_i^{\mathrm{geo}}\right), \mathrm{Mean}_i\left(w_i\right), \mathrm{Mean}_i\left(m_i\right)\right]. \tag{5}$$

This object-level repair context is provided to both the global and local velocity fields. The local field additionally receives the token-wise repair-state set $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^M$, where the reliability component guides neighborhood selection and attention, while the complete repair state modulates local features through point-wise FiLM. Both $\mathbf{c}_g$ and $\mathcal{C}$ are computed once from the GALR output and held fixed throughout latent transport.

Global Flow Matching. During training, the clean target is encoded as $(Z_g^{\mathrm{c}}, Z_\ell^{\mathrm{c}}) = E(\boldsymbol{X}_{\mathrm{c}})$. For $t \sim \mathcal{U}(0,1)$, we use a linear interpolation and its corresponding target velocity (Lipman et al., 2023)

$$Z_t^g = (1-t)Z_g^{\mathrm{n}} + tZ_g^{\mathrm{c}}, \qquad U_g = Z_g^{\mathrm{c}} - Z_g^{\mathrm{n}}. \tag{6}$$

The global velocity field $v_\theta^g(Z_t^g, t \mid c_g)$ is parameterized by an MLP and trained to predict U_g. The global branch also provides a time-dependent condition for local restoration.

Globally Conditioned Local Flow Matching. Conditioned on the global velocity state, the local branch restores geometry and latent features from the repaired initialization $\widetilde{\boldsymbol{Z}}_\ell^{\mathrm{n}}$. We first establish local training correspondence:

$$\boldsymbol{Z}_\ell^{\mathrm{tar}} = \mathcal{A}\left(\boldsymbol{Z}_\ell^{\mathrm{c}}; \widetilde{\boldsymbol{Z}}_\ell^{\mathrm{n}}\right), \tag{7}$$

where $\mathcal{A}$ assigns each repaired anchor the complete clean token associated with its nearest clean anchor in Euclidean space. Using the same sampled time t as the global branch, we define

$$\boldsymbol{Z}_t^\ell = (1-t)\widetilde{\boldsymbol{Z}}_\ell^{\mathrm{n}} + t\boldsymbol{Z}_\ell^{\mathrm{tar}}, \qquad \boldsymbol{U}_\ell = \boldsymbol{Z}_\ell^{\mathrm{tar}} - \widetilde{\boldsymbol{Z}}_\ell^{\mathrm{n}}. \tag{8}$$

A four-layer KNN Transformer parameterizes $\boldsymbol{v}_\theta^\ell(\boldsymbol{Z}_t^\ell, t \mid \boldsymbol{Z}_t^g, \boldsymbol{c}_g, \mathcal{C})$ and predicts $\boldsymbol{U}_\ell$. The global state supplies object-level context at each transport time, while the fixed repair states identify the

reliability and prior correction of individual tokens. For evolving anchors $\boldsymbol{x}_i(t)$, candidate-neighbor reliability modifies neighborhood selection and attention:

$$\widetilde{d}_{ij}(t) = \|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)\|_2 + \lambda_{\mathrm{knn}}(1-t)(1-r_j), \qquad b_{ij}^{\mathrm{rel}} = \log(r_j + \epsilon). \tag{9}$$

Here, $x_i(t)$ and $x_j(t)$ denote the current spatial anchors of the center token i and candidate neighbor j, respectively, while r_j is the reliability of neighbor and λ_{knn} controls the reliability penalty. The adjusted distance $\widetilde{d}_{ij}(t)$ guides KNN selection, whereas b_{ij}^{rel} biases attention against unreliable neighbors. Neighborhoods are recomputed during transport, with the selection penalty gradually reduced by $(1-t)$.

Global conditioning enters each Transformer sublayer through shared AdaLN parameters $(\boldsymbol{\beta}_g^{(s)}, \boldsymbol{\gamma}_g^{(s)})$ predicted from $(\boldsymbol{Z}_t^g, t, \boldsymbol{c}_g)$. Token-wise repair states provide additional FiLM parameters:

$$\left(\beta_{p,i}^{(s)}, \gamma_{p,i}^{(s)}\right) = f_{\mathrm{FiLM}}^{(s)}(c_i), \qquad \widetilde{h}_i^{(s)} = \left(1 + \gamma_g^{(s)} + \gamma_{p,i}^{(s)}\right) \odot \mathrm{LN}\left(h_i^{(s)}\right) + \beta_g^{(s)} + \beta_{p,i}^{(s)}. \tag{10}$$

Thus, the current global state modulates all local tokens, while the repair states adjust processing at individual tokens. Here, LN denotes layer normalization and $\odot$ denotes element-wise multiplication.

The linear paths define training supervision. At inference, the local field instead receives the current state of the learned global trajectory:

$$\begin{aligned} \frac{\mathrm{d}Z_t^g}{\mathrm{d}t} &= v_\theta^g\left(Z_t^g, t \mid c_g\right), & Z_0^g &= Z_g^{\mathrm{n}}, \\ \frac{\mathrm{d}Z_t^\ell}{\mathrm{d}t} &= v_\theta^\ell\left(Z_t^\ell, t \mid Z_t^g, c_g, C\right), & Z_0^\ell &= \widetilde{Z}_\ell^{\mathrm{n}}. \end{aligned} \tag{11}$$

The two fields are integrated over the same interval, so local restoration remains conditioned on the evolving global representation throughout inference. At $t = 1$, the restored endpoints $\widehat{\boldsymbol{Z}}_g = \boldsymbol{Z}_g(1)$ and $\widehat{\boldsymbol{Z}}_\ell = \boldsymbol{Z}_\ell(1)$ are jointly decoded as $\boldsymbol{X}_{\mathrm{p}} = D(\widehat{\boldsymbol{Z}}_g, \widehat{\boldsymbol{Z}}_\ell)$.

4.4 Learning Objectives

Training proceeds in two stages. We first optimize GALR, then freeze it while training the global and local velocity fields. The second stage therefore uses fixed repaired initializations and repair conditions for each observation.

GALR Objective. GALR is supervised by geometric and reliability objectives:

$$\mathcal{L}_{\mathrm{GALR}} = \mathcal{L}_{\mathrm{set}} + \lambda_{\mathrm{cal}}\mathcal{L}_{\mathrm{cal}}, \tag{12}$$

where $\mathcal{L}_{\mathrm{set}}$ is a bidirectional anchor-set loss on the repaired geometry and $\mathcal{L}_{\mathrm{cal}}$ is a focal loss for corruption-score estimation. With GALR frozen, we optimize

GLFM Objective. After GALR is frozen, the Flow Matching stage is optimized with

$$\mathcal{L}_{\mathrm{GLFM}} = \lambda_{\mathrm{FM}}\mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{lat}}\mathcal{L}_{\mathrm{lat}} + \lambda_{\mathrm{dec}}\mathcal{L}_{\mathrm{dec}}, \tag{13}$$

where the hierarchical Flow Matching objective is

$$\mathcal{L}_{\mathrm{FM}} = \lambda_g\mathcal{L}_{\mathrm{FM}}^g + \lambda_\ell\mathcal{L}_{\mathrm{FM}}^\ell. \tag{14}$$

The global and local Flow Matching terms regress the predicted velocity fields toward the target velocities defined in Sec. 4.3. The endpoint losses $\mathcal{L}_{\mathrm{lat}}$ and $\mathcal{L}_{\mathrm{dec}}$ supervise the restored restoration in latent and decoded point-cloud spaces, respectively. Complete loss definitions and weighting coefficients are provided in the Appendix.

5 Experiments

5.1 Experimental Setup

Dataset. Following the protocal of SPORE (Sec. 3), we evaluate all methods on SPORE's Low, Mid, and High benchmark datasets, reporting results separately for each corruption regime.

Baselines. We compare PRISM with nine representative methods spanning statistical filtering, discriminative correction, and generative restoration: Statistical Outlier Removal (SOR) (Rusu & Cousins, 2011), PointCleanNet (PCN) (Rakotosaona et al., 2020), Score-Based Point Cloud Denoising (Luo & Hu, 2021b), PD-Flow (Mao et al., 2022), P2P Bridge (Vogel et al., 2024), IterativePFN (de Silva Edirimuni et al., 2023), PD-LTS (Mao et al., 2024), ASDN (Guo et al., 2025), and PGD (Guo et al., 2026). All learning-based baselines are adapted to SPORE and retrained on the same training split.

Evaluation metrics. We report four complementary metrics: CD-L2 (Fan et al., 2017) for point-set fidelity, P2M (Ravi et al., 2020) for reference-surface consistency, F@0.01 (Tatarchenko et al., 2019) for correspondence and coverage at a distance threshold of 0.01, and HD95 (Huttenlocher et al., 1993) for the 95th percentile of bidirectional distances. Detailed definitions are provided in the Appendix.

Implementation details. All experiments use a single NVIDIA RTX PRO 6000 GPU. We first train the hierarchical autoencoder on clean point clouds and freeze it, then train GALR for 50 epochs with a batch size of 64 and a learning rate of 10^{-5}. With GALR frozen, we train GLFM for 80 epochs using a micro-batch size of 32 and four-step gradient accumulation (effective batch size: 128). Its learning-rate schedule uses 5 epochs of linear warmup followed by cosine decay from 5×10^{-5} to 5×10^{-6}. Both restoration stages use AdamW with weight decay 10^{-4}. The local branch uses $K = 16$ neighbors, and inference uses a four-step Euler solver.

5.2 Comparison Study

Table 1 compares restoration performance across three single-photon degradation regimes. Under Low degradation, rankings vary across metrics: PGD, PD-LTS, and PD-Flow achieve lower CD-L2 than PRISM, while PGD and PD-LTS also obtain lower P2M. PRISM achieves the lowest HD95 (0.0279) and ties SOR for the highest F@0.01 (0.1997), demonstrating advantages in threshold-based correspondence and upper-tail error despite higher average geometric discrepancies. Under Mid degradation, PRISM leads on CD-L2, F@0.01, and HD95, while its P2M (1.242) remains close to the best result (1.200).

Under High degradation, PRISM ranks first on all four metrics. Relative to the strongest competing result for each metric, it reduces CD-L2, P2M, and HD95 by 2.4%, 18.2%, and 27.6%, respectively, and increases F@0.01 by 22.9%. The larger gains in F@0.01 and HD95 indicate improvements in

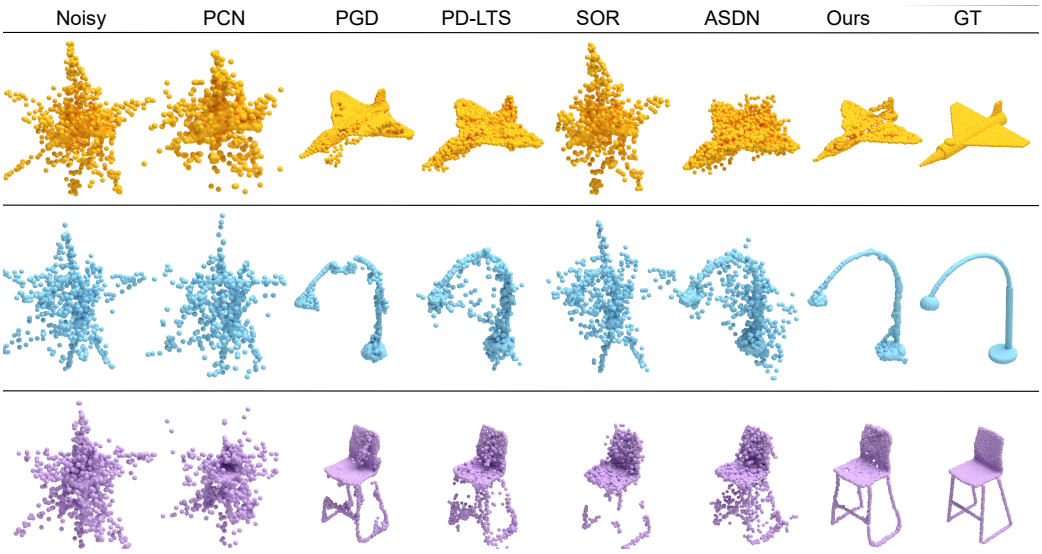


Figure 3: Qualitative comparison under High single-photon degradation. We compare the noisy observation with selected competitive baselines, including PCN, PGD, PD-LTS, SOR, and ASDN. Our method suppresses severe spatial outliers while preserving coherent object geometry and thin structures, yielding restorations that more closely match the ground truth (GT).

Table 1: Quantitative comparison under Low, Mid, and High single-photon degradation. CD-L2 and P2M are reported in units of 10^{-3}.

Model	Low noise				Mid noise				High noise			
	CD-L2↓	P2M↓	F@0.01↑	HD95↓	CD-L2↓	P2M↓	F@0.01↑	HD95↓	CD-L2↓	P2M↓	F@0.01↑	HD95↓
Noisy	3.936	3.612	0.1993	0.0390	32.813	32.107	0.1246	0.0493	148.670	146.872	0.0559	0.6862
PD-Flow	1.115	1.383	0.1306	0.0343	2.288	1.885	0.1200	0.0373	17.007	4.528	0.2781	0.0410
Score-based	5.859	5.890	0.0324	0.0857	8.859	8.586	0.0240	0.1187	39.635	15.674	0.0468	0.2608
PGD	**0.933**	**0.989**	0.1724	0.0321	1.190	**1.200**	0.1358	0.0346	5.393	2.632	0.0900	0.0489
P2P-Bridge	12.320	6.496	0.0321	0.1464	12.586	7.175	0.0308	0.1487	15.419	12.559	0.0233	0.1697
SOR	1.176	1.467	**0.1997**	0.0388	2.488	2.020	0.1279	0.0436	86.140	84.753	0.0585	0.4509
PCN	4.151	2.836	0.1856	0.0492	30.933	7.638	0.1212	0.0954	109.644	17.590	0.2293	0.8116
ASDN	2.158	1.966	0.1883	0.0397	4.023	3.409	0.1167	0.0462	8.052	7.307	0.0593	0.1130
IterativePFN	2.107	2.455	0.1569	0.0514	5.126	3.572	0.0788	0.0930	34.944	9.675	0.0409	0.2292
PD-LTS	0.987	1.104	0.1791	0.0330	1.215	1.201	0.1314	0.0370	5.376	3.545	0.0770	0.0547
Ours	1.164	1.240	**0.1997**	**0.0279**	**1.174**	1.242	**0.2143**	**0.0237**	**5.248**	**2.154**	**0.3419**	**0.0297**

threshold-based correspondence and upper-tail error that are less apparent from average point-set discrepancy alone. For example, PGD and PD-LTS attain CD-L2 values close to PRISM but exhibit substantially lower F@0.01 and higher HD95. Conversely, PCN's relatively high F@0.01 (0.2293) coexists with large CD-L2 (109.644) and HD95 (0.8116), showing that favorable correspondence within a fixed threshold can accompany substantial residual errors. PRISM's simultaneous gains across all four metrics support its ability to recover surface geometry while controlling large residual deviations under severe corruption.

Figure 3 illustrates these differences under High degradation. PCN and SOR retain substantial isolated corruption, while ASDN removes some outliers but leaves incomplete local geometry. PGD and PD-LTS recover the dominant shape more effectively, although thin or connected regions remain fragmented in several examples. PRISM suppresses visible outliers and produces more continuous geometry, particularly along the curved thin structure in the second example and the chair legs and supporting bars in the third. The chair illustrates the challenge of recovering both narrow components and their connections; their greater continuity in PRISM's reconstruction provides qualitative evidence of improved recovery in structurally intricate regions. These observations are consistent with GALR's selective correction of unreliable anchors and GLFM's integration of retained repair information with global shape context, complementing the quantitative improvements in correspondence and residual-error control.

5.3 Real-World Results.

We apply the model trained on SPORE to real single-photon point clouds acquired using a single-photon avalanche diode (SPAD) detector and a two-axis scanning galvanometer. Figure 4 shows the

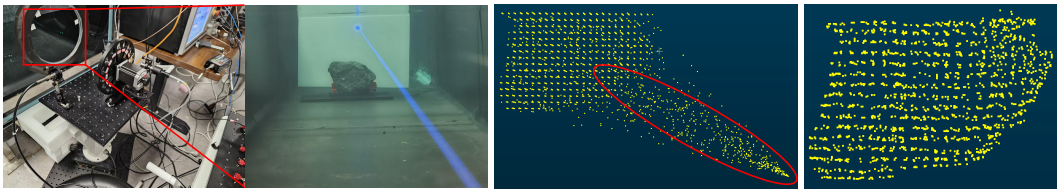


Figure 4: Our single-photon LiDAR, the collected noisy point cloud and its corresponding restored point cloud.

acquisition system, corrupted observation, and PRISM reconstruction. The capture approximates the single-frame acquisition setting modeled by SPORE. PRISM visibly suppresses corruption and produces a smoother, more coherent reconstruction, providing qualitative evidence of transfer from simulated training data to real measurements.

5.4 Ablation Study

We evaluate the proposed components under all three single-photon degradation regimes. For compact presentation, Table 2 reports the High regime, which directly probes severe corruption; the corresponding Low and Mid results are provided in Appendix G. The ablations cover framework-level components and internal GLFM design choices.

Table 2: Hierarchical ablation study under High single-photon degradation. CD and P2M are reported in units of 10^{-3}. The upper block compares framework-level configurations, while the lower block individually ablates internal Flow Matching designs from the full framework.

Framework			Flow design		Metrics			
GALR	GLFM	Repair State	Reliability-aware Neighborhood	FiLM	CD↓	P2M↓	F@.01↑	HD95↓
Framework-level ablations								
✓					33.999	10.017	0.0573	0.0326
	✓		✓	✓	18.130	15.018	0.0778	0.0829
✓	✓		✓	✓	6.089	2.848	0.0718	0.0618
✓	✓	✓	✓	✓	**5.248**	**2.154**	**0.3419**	**0.0297**
Flow Matching design ablations								
✓	✓	✓		✓	6.831	3.020	0.0545	0.0722
✓	✓	✓	✓		5.496	**2.154**	0.2404	0.0568
✓	✓	✓	✓	✓	**5.248**	**2.154**	**0.3419**	**0.0297**

Framework-level ablations. Removing GALR degrades all four metrics, supporting its framework-level contribution under severe corruption. Because this intervention removes both GALR-based pre-repair and GALR-derived conditioning, it does not isolate repaired initialization alone. Conversely, GALR without GLFM remains substantially behind the full model in CD-L2, P2M, and F@0.01, although its HD95 is already close to the full result. These results support complementary roles for GALR and GLFM: GALR suppresses severe local corruption, while GLFM further restores the hierarchical representation. The repair-conditioning ablation separates repaired initialization from persistent conditioning. GLFM starts from the same repaired local latents as the full model, but the GALR-derived object-level context c_g and token-wise repair-state set C are removed during transport. CD-L2 increases from 5.248 to 6.089, P2M from 2.154 to 2.848, F@0.01 decreases from 0.3419 to 0.0718, and HD95 increases from 0.0297 to 0.0618. Thus, repaired initialization alone does not explain the full gain; persistent GALR-derived conditioning provides additional guidance during latent transport.

GLFM design ablations. Removing reliability-aware neighborhood interaction degrades all four metrics: CD-L2 increases from 5.248 to 6.831, P2M from 2.154 to 3.020, F@0.01 decreases from 0.3419 to 0.0545, and HD95 increases from 0.0297 to 0.0722. This result supports the importance of using token reliability in neighborhood construction and attention under severe corruption. Removing point-wise FiLM has a smaller effect on the average-distance metrics: CD-L2 increases to 5.496, while P2M remains unchanged at the reported precision. In contrast, F@0.01 decreases to 0.2404 and HD95 increases to 0.0568. Point-wise FiLM therefore contributes most clearly to strict local correspondence and control of large residual geometric errors. Overall, the ablations support a consistent information pathway. GALR improves the initial latent geometry and produces repair information that is retained throughout GLFM, while reliability-aware neighborhood interaction and point-wise FiLM provide two local mechanisms for exploiting this information during restoration.

6 Conclusion

We introduced PRISM and SPORE, a restoration framework and physics-grounded simulated benchmark for structured single-photon corruption. PRISM selectively repairs unreliable latent anchors using reliable neighboring geometry and retains the resulting repair information to condition global-local flow matching. Experiments show competitive performance under mild corruption and the best results among evaluated methods on all four metrics with complex structures under High corruption. Ablations support the complementary benefits of pre-repair, persistent repair-state conditioning, reliability-aware neighborhood interactions, and point-wise FiLM. These findings highlight the value of preserving reliability and correction information when geometric evidence is severely degraded. Future work will further evaluate generalization to real single-photon acquisition conditions and other forms of structured sensor degradation.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on author-developed research methodology and experimental design, and to assist with the presentation of experimental analyses and interpretations developed by the authors. We did not use generative AI tools for other applicable tasks requiring disclosure under the ICLR policy. Additionally, we used generative AI tools for literature search, manuscript structuring, language polishing, and suggestions for scientific figure design. All AI-assisted suggestions and content were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The source code and SPORE benchmark will be made publicly available. Section 3 describes the benchmark construction and data splits, while Section 5.1 summarizes the main training, hardware, and inference settings. The Appendix provides the remaining reproducibility details, including the single-photon simulation and benchmark validation (Appendices A–B), exact evaluation metrics (Appendix C), detailed GALR and GLFM architectures (Appendix D), complete learning objectives and training procedures (Appendix E), and baseline adaptation and retraining protocols (Appendix F).

REFERENCES

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

Xiaoqian Cheng, Dong Xiao, Husen Li, Zheng Liu, and Renjie Chen. Routing on demand: DSNet for efficient progressive point cloud denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39111–39120, 2026.

Dasith de Silva Edirimuni, Xuequan Lu, Zhiwen Shao, Gang Li, Antonio Robles-Kelly, and Ying He. IterativePFN: True iterative point cloud filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13530–13539, 2023.

Dasith de Silva Edirimuni, Xuequan Lu, Gang Li, Lei Wei, Antonio Robles-Kelly, and Hongdong Li. StraightPCF: Straight point cloud filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20721–20730, 2024.

Peng Du, Xingce Wang, Zhongke Wu, Xudong Ru, Xavier Granier, and Ying He. LaPDA: Latent-space point cloud denoising with adaptivity. *IEEE Transactions on Visualization and Computer Graphics*, 32(2):1525–1539, 2026. doi: 10.1109/TVCG.2025.3635138.

Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2463–2471, 2017. doi: 10.1109/CVPR.2017.264.

Bhavya Goyal, Felipe Gutierrez-Barragan, Wei Lin, Andreas Velten, Yin Li, and Mohit Gupta. Robust 3D object detection using probabilistic point clouds from single-photon LiDARs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28417–28427, 2025.

Chuchen Guo, Weijie Zhou, Zheng Liu, and Ying He. You should learn to stop denoising on point clouds in advance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3212–3219, 2025. doi: 10.1609/aaai.v39i3.32331.

Chuchen Guo, Zheng Liu, and Ying He. Guiding point cloud denoising with learned structural priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4395–4402, 2026. doi: 10.1609/aaai.v40i6.42437.

Abderrahim Halimi, Jean-Yves Tourneret, Aongus McCarthy, Ewan Wade, Rachael Tobin, Jorge Garcia-Armenta, Phil Soan, and Gerald S. Buller. Continuous-surface 3D reconstruction from kilometer-range single-photon LiDAR using score-based priors. *Scientific Reports*, 2026. doi: 10.1038/s41598-026-57483-5. Advance online publication.

Ka-Hei Hui, Chao Liu, Xiaohui Zeng, Chi-Wing Fu, and Arash Vahdat. Not-so-optimal transport flows for 3D point cloud generation. In *International Conference on Learning Representations (ICLR)*, 2025.

Daniel P. Huttenlocher, Gregory A. Klanderman, and William J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993. doi: 10.1109/34.232073.

Ahmed Kirmani, Dheera Venkatraman, Dongeek Shin, Andrea Colaço, Franco N. C. Wong, Jeffrey H. Shapiro, and Vivek K. Goyal. First-photon imaging. *Science*, 343(6166):58–61, 2014. doi: 10.1126/science.1246775.

Jongho Lee, Atul Ingle, Jenu V. Chacko, Kevin W. Eliceiri, and Mohit Gupta. CASPI: Collaborative photon processing for active single-photon imaging. *Nature Communications*, 14:3158, 2023. doi: 10.1038/s41467-023-38893-9.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Zheng Liu, Yaowu Zhao, Sijing Zhan, Yuanyuan Liu, Renjie Chen, and Ying He. PCDNF: Revisiting learning-based point cloud denoising via joint normal filtering. *IEEE Transactions on Visualization and Computer Graphics*, 30(8):5419–5436, 2024. doi: 10.1109/TVCG.2023.3292464.

Zheng Liu, Weijie Zhou, Chuchen Guo, Qinjun Qiu, and Zhong Xie. PyramidPCD: A novel pyramid network for point cloud denoising. *Pattern Recognition*, 161:111228, 2025. doi: 10.1016/j.patcog.2024.111228.

Zheng Liu, Zhenyu Huang, Maodong Pan, and Ying He. Deterministic point cloud diffusion for denoising. *IEEE Transactions on Visualization and Computer Graphics*, 32(2):1822–1834, 2026a. doi: 10.1109/TVCG.2025.3621633.

Zhi-Song Liu, Chenhang He, Yakun Ju, and Lei Li. PUFM: Efficient point cloud upsampling via flow matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 7458–7466, 2026b. doi: 10.1609/aaai.v40i9.37685.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3D point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2837–2845, 2021a.

Shitong Luo and Wei Hu. Score-based point cloud denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4583–4592, 2021b.

Weihan Luo, Anagh Malik, and David B Lindell. Transientangelo: Few-viewpoint surface reconstruction using single-photon lidar. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 8723–8733. IEEE, 2025. doi: 10.1109/WACV61041.2025.00845.

Aihua Mao, Zihui Du, Yu-Hui Wen, Jun Xuan, and Yong-Jin Liu. PD-Flow: A point cloud denoising framework with normalizing flows. In *European Conference on Computer Vision (ECCV)*, pp. 398–415, 2022.

Aihua Mao, Biao Yan, Zijing Ma, and Ying He. Denoising point clouds in latent space via graph convolution and invertible neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5768–5777, 2024. doi: 10.1109/CVPR52733.2024.00551.

Aongus McCarthy, Gregor G. Taylor, Jorge Garcia-Armenta, Boris Korzh, Dmitry V. Morozov, Andrew D. Beyer, Ryan M. Briggs, Jason P. Allmaras, Bruce Bumble, Marco Colangelo, Di Zhu, Karl K. Berggren, Matthew D. Shaw, Robert H. Hadfield, and Gerald S. Buller. High-resolution long-distance depth imaging LiDAR with ultra-low timing jitter superconducting nanowire single-photon detectors. *Optica*, 12(2):168–177, 2025. doi: 10.1364/OPTICA.544877.

Fangzhou Mu, Carter Sifferman, Sacha Jungerman, Yiquan Li, Mark Han, Michael Gleicher, Mohit Gupta, and Yin Li. Towards 3D vision with low-cost single-photon cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5302–5311, 2024.

Tomoki Ohno, Yoshiyuki Utagawa, Masashi Tanabe, Tsubasa Sugiyama, and Tomoki Matsumura. Super-dense point clouds acquired by an ultralight 10 g solid-state single photon lidar. *Nature Communications*, 16(1):11567, 2025. doi: 10.1038/s41467-025-67346-8.

Marie-Julie Rakotosaona, Vittorio La Barbera, Paul Guerrero, Niloy J. Mitra, and Maks Ovsjanikov. PointCleanNet: Learning to denoise and remove outliers from dense point clouds. *Computer Graphics Forum*, 39(1):185–203, 2020. doi: 10.1111/cgf.13753.

Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020.

Radu Bogdan Rusu and Steve Cousins. 3D is here: Point cloud library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011. IEEE. doi: 10.1109/ICRA.2011.5980567.

Dongeek Shin, Ahmed Kirmani, Vivek K. Goyal, and Jeffrey H. Shapiro. Photon-efficient computational 3-D and reflectivity imaging with single-photon detectors. *IEEE Transactions on Computational Imaging*, 1(2):112–125, 2015. doi: 10.1109/TCI.2015.2453093.

Julián Tachella, Yoann Altmann, Nicolas Mellado, Aongus McCarthy, Rachael Tobin, Gerald S. Buller, Jean-Yves Tourneret, and Stephen McLaughlin. Real-time 3D reconstruction from single-photon LiDAR data using plug-and-play point cloud denoisers. *Nature Communications*, 10: 4984, 2019. doi: 10.1038/s41467-019-12943-7.

Maxim Tatarchenko, Stephan R. Richter, Rene Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3405–3414, 2019. doi: 10.1109/CVPR.2019.00352.

Mathias Vogel, Keisuke Tateno, Marc Pollefeys, Federico Tombari, Marie-Julie Rakotosaona, and Francis Engelmann. P2P-Bridge: Diffusion bridges for 3D point cloud denoising. In *European Conference on Computer Vision (ECCV)*, pp. 184–201, 2024.

Martin Weinmann, Boris Jutzi, Stefan Hinz, and Clément Mallet. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of photogrammetry and remote sensing*, 105:286–304, 2015.

Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. PointFlow: 3D point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4541–4550, 2019. doi: 10.1109/ICCV.2019.00464.

Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. LION: Latent point diffusion models for 3D shape generation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 10021–10039, 2022.

Dongbo Zhang, Xuequan Lu, Hong Qin, and Ying He. Pointfilter: Point cloud filtering via encoder-decoder modeling. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2015–2027, 2021. doi: 10.1109/TVCG.2020.3027069.

Qingyuan Zhou, Weidong Yang, Ben Fei, Jingyi Xu, Rui Zhang, Keyi Liu, Yeqi Luo, and Ying He. 3DMambaIPF: A state space model for iterative point cloud filtering via differentiable rendering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10843–10851, 2025. doi: 10.1609/aaai.v39i10.33178.

A PHYSICS-GROUNDED SINGLE-PHOTON SIMULATION DETAILS

We detail the physics-grounded single-photon simulator introduced in Sec. 3.2. The simulator converts each normalized triangular mesh into paired clean and corrupted point clouds. Multi-view ray casting first determines visible surface geometry. Signal, background, and dark-count photons are then sampled using Poisson statistics. Finally, stochastic detection, spurious returns, timing uncertainty, range perturbation, and angular jitter produce the corrupted observation.

The clean and corrupted branches share the same candidate ray geometry. The clean reference contains valid ray–mesh intersections before detector corruption, whereas the corrupted branch applies the photon-detection and measurement model to the same candidate returns. Both variable-size point sets are then converted to the 2,048-point representations used for restoration.

A.1 ACQUISITION GEOMETRY AND VISIBLE-SURFACE CONSTRUCTION

Each ShapeNet mesh is centered and normalized before ray casting. Let the original mesh vertices be $\mathcal{V} = \{\mathbf{v}_k\}_{k=1}^{N_v}$. We compute the midpoint of the axis-aligned bounding box, $\mathbf{c} = \frac{\mathbf{v}_{\min}+\mathbf{v}_{\max}}{2}$, and normalize the centered vertices using their maximum radial distance,

$$s = \max_k \|\mathbf{v}_k - \mathbf{c}\|_2 , \qquad \widetilde{\mathbf{v}}_k = \frac{\mathbf{v}_k - \mathbf{c}}{s}. \tag{15}$$

This operation centers the object at the origin and sets its maximum vertex radius to one while preserving its aspect ratio.

Scale convention. ShapeNet meshes do not provide a shared object-level metric scale. We therefore use the normalized coordinates as a nominal acquisition coordinate system. The normalized ray range enters the inverse-square signal model directly, and timing-derived and configured range perturbations are applied numerically in the same coordinate system. Accordingly, parameters expressed in metric units define the nominal scale of the simulation. This convention maintains consistent coupling among propagation range, photon statistics, and measurement uncertainty across object categories.

We use the full-body acquisition mode implemented in the simulator. Ten sensor viewpoints are distributed approximately uniformly on a sphere of radius 2.2 using Fibonacci-sphere sampling. Each sensor is oriented toward the origin. A 96×96 perspective ray grid is generated at each viewpoint with a field of view of $45°$.

For a pixel (u, v), the camera-frame ray coordinates are

$$\begin{aligned} x_{uv} &= \left(2\frac{u+0.5}{W} - 1\right)\tan\frac{\theta_{\text{fov}}}{2}\frac{W}{H}, \\ y_{uv} &= \left(1 - 2\frac{v+0.5}{H}\right)\tan\frac{\theta_{\text{fov}}}{2}. \end{aligned} \tag{16}$$

Here, $W = H = 96$. The camera-frame direction is normalized and transformed into the world frame using the view-specific look-at rotation.

For each ray, $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}, \quad t > 0$, we compute its intersections with the mesh triangles. If multiple intersections are present, only the nearest valid positive intersection is retained, $t^\star = \min\{t > 0 : \mathbf{o} + t\mathbf{d} \in \mathcal{M}\}$. The corresponding visible surface point is $\mathbf{x}_{\text{vis}} = \mathbf{o} + t^\star\mathbf{d}$. Only the first visible surface along each ray contributes to the clean point cloud, so occluded intersections are excluded. Visible intersections from all ten viewpoints are concatenated to form the raw clean reference.

A.2 SIGNAL-PHOTON FORMATION

For each valid ray–surface intersection, the expected number of detected signal photons depends on the emitted photon level and the local acquisition geometry. A photon at wavelength λ_0 has energy $E_\gamma = \frac{hc}{\lambda_0}$, where h is Planck's constant and c is the speed of light. Given pulse energy E_{pulse} and N_{pulse} emitted pulses, the emitted photon count is $N_{\text{emit}} = \frac{E_{\text{pulse}}}{E_\gamma}N_{\text{pulse}}$. For a surface

intersection at range d_i, the simulator computes the incidence term as $\mu_i = \left|-\mathbf{d}_i^\top \mathbf{n}_i\right|$, where $\mathbf{d}_i$ is the incident ray direction and $\mathbf{n}_i$ is the corresponding triangle normal. Surface reflectivity is sampled independently for each candidate return, $\rho_i \sim \text{clip}\left[\mathcal{N}(\mu_\rho, \sigma_\rho^2), 0.02, 0.98\right]$, where the mean reflectivity is fixed at $\mu_\rho = 0.55$, while σ_ρ varies across corruption regimes.

The expected detected signal count is

$$\lambda_{s,i} = \text{clip}\left[\kappa_{\text{sim}} N_{\text{emit}} \rho_i \frac{\mu_i}{\max(d_i^2, 10^{-6})} \eta_{\text{det}} \eta_{\text{opt}} \eta_{\text{atm}}, 0, \lambda_{\max}\right], \tag{17}$$

where η_{det}, η_{opt}, and η_{atm} denote detector, optical, and atmospheric efficiencies, respectively. The current implementation uses $\kappa_{\text{sim}} = 10^{-6}$ and $\lambda_{\max} = 50$. The signal count is then sampled as $N_{s,i} \sim \text{Poisson}(\lambda_{s,i})$. Consequently, the probability of observing a surface return varies with range, incidence angle, reflectivity, emitted photon level, and sensor efficiency.

A.3 Background Photons, Dark Counts, and Stochastic Detection

Background photons and detector dark counts are modeled as an independent Poisson component. For each valid ray–surface candidate, $\lambda_b = \lambda_{\text{bg}} + \lambda_{\text{dark}}, \quad N_b \sim \text{Poisson}(\lambda_b)$. For a Poisson process with rate λ, the probability of observing at least one count is $P(N \geq 1) = 1 - \exp(-\lambda)$. A signal return is first considered detected when $N_s \geq 1$. We then apply an incidence-dependent dropout probability, $p_{\text{drop},i} = \text{clip}\left[p_0 + 0.15(1 - \mu_i), 0, 0.95\right]$, where p_0 is the regime-specific base dropout rate. A signal measurement is retained only if it survives this stochastic dropout. The final detection condition is $I_{\text{det}} = I_{\text{signal}} \vee \mathbb{I}(N_b \geq 1)$. A valid ray can therefore produce a retained surface measurement, a background-induced measurement, or no measurement.

Background photons and dark counts are sampled on rays with valid mesh intersections. SPORE therefore models background-induced range corruption within the candidate object-ray support used to form the object-level observation.

A.4 Background-Induced and Spurious Returns

For a retained signal measurement, the initial range is the physical ray–surface intersection range $t^\star$. If no signal survives but one or more background photons are observed, the surface range is discarded and replaced by $d_{\text{bg}} \sim \mathcal{U}(0, d_{\text{far}}), \quad d_{\text{far}} = 3.2$. The simulator additionally applies an independent spurious-return process, $I_{\text{spur}} \sim \text{Bernoulli}(p_{\text{spur}})$. If a retained measurement is selected as spurious, its current range is replaced by $d_{\text{spur}} \sim \mathcal{U}(0, d_{\text{far}})$. Background-only and selected spurious measurements are assigned ranges sampled from $\mathcal{U}(0, d_{\text{far}})$.

A.5 Timing, Range, and Angular Uncertainty

Retained measurements are further perturbed by timing and range uncertainty. A timing jitter of σ_t is converted to one-way range uncertainty as $\sigma_{\text{time}} = \frac{c\sigma_t}{2}$. The simulator combines this term with the configured range perturbation, $\sigma_r = \sqrt{\sigma_{\text{time}}^2 + \sigma_{\text{range}}^2}$. If d_0 denotes the current range after background and spurious-return processing, the noisy range is $\widetilde{d} = \max(0, d_0 + \epsilon_r), \quad \epsilon_r \sim \mathcal{N}(0, \sigma_r^2)$.

Angular uncertainty is applied independently. For each ray direction $\mathbf{d}$, a random tangent-plane unit vector $\mathbf{q}$ is constructed such that $\mathbf{q}^\top \mathbf{d} = 0$. The perturbed direction is

$$\widetilde{\mathbf{d}} = \frac{\mathbf{d} + \epsilon_\theta \mathbf{q}}{\|\mathbf{d} + \epsilon_\theta \mathbf{q}\|_2}, \qquad \epsilon_\theta \sim \mathcal{N}(0, \sigma_\theta^2). \tag{18}$$

The corrupted 3D detection is finally reconstructed as $\mathbf{x}_{\text{n}} = \mathbf{o} + \widetilde{d}\,\widetilde{\mathbf{d}}$. Range and angular perturbations are applied after stochastic detection and range replacement. Both background and signal detections therefore pass through the same final measurement model.

A.6 Low, Mid, and High Corruption Regimes

The three corruption regimes use the same acquisition geometry and simulation procedure. They differ only in a predefined set of photon and measurement parameters, summarized in Table 3. Fixed

parameters include the number and arrangement of views, wavelength, detector efficiency, optical efficiency, atmospheric efficiency, mean reflectivity, and photon-rate scaling. Increasing corruption simultaneously lowers pulse energy and increases background counts, dark counts, dropout, reflectivity variation, timing jitter, angular jitter, range noise, and spurious detections.

Low, Mid, and High are generated by coordinated changes in these acquisition factors, which jointly control return availability and measurement perturbation.

A.7 Fixed-Size Point-Cloud Construction

The visible intersections from all views are concatenated into a variable-size clean point cloud, $\mathbf{X}_{\text{raw}}^{\text{c}} = \bigcup_{v=1}^{V} \mathbf{X}_{v}^{\text{c}}, \quad V = 10$, and the detected points are concatenated similarly, $\mathbf{X}_{\text{raw}}^{\text{n}} = \bigcup_{v=1}^{V} \mathbf{X}_{v}^{\text{n}}$. The two clouds can contain different numbers of points because photon detection and dropout are stochastic.

For the clean reference, the preprocessing script applies farthest-point sampling, $\mathbf{X}_{\text{c}} = \text{FPS}_{2048}\left(\mathbf{X}_{\text{raw}}^{\text{c}}\right)$. For the corrupted observation, the reported benchmark uses the fixed radius-2.0 configuration,

$$\mathbf{X}_{\text{crop}}^{\text{n}} = \left\{\mathbf{x} \in \mathbf{X}_{\text{raw}}^{\text{n}} : \|\mathbf{x}\|_2 \leq 2.0\right\}, \tag{19}$$

followed by random sampling without replacement, $\mathbf{X}_{\text{n}} = \text{RandomSample}_{2048}\left(\mathbf{X}_{\text{crop}}^{\text{n}}\right)$. The clean and corrupted branches deliberately use different fixed-size sampling rules: farthest-point sampling produces a coverage-oriented clean reference, whereas random sampling preserves the acquisition-induced non-uniformity of the corrupted observation. If either the raw clean cloud or the radius-cropped corrupted cloud contains fewer than 2,048 points, that noisy–clean pair is excluded from the fixed-size benchmark. The resulting regime-specific pair counts are reported in Appendix B.1.

In summary, the simulator proceeds from mesh normalization to multi-view ray casting, visible-surface intersection, photon counting, stochastic detection, background or spurious range replacement, measurement perturbation, and fixed-size point-cloud construction.

A.8 Randomization and Reproducibility

The simulator uses a base random seed of 1234. Input OBJ files are processed in sorted path order. For object index i and corruption-level index ℓ, the object/regime seed is $s_{i,\ell} = 1234 + 100000\, i + 1009\, \ell$, modulo 2^{32}. During full-body acquisition, the v-th viewpoint uses the additional offset $+v$. The seed is propagated to Python, NumPy, and PyTorch random-number generators before stochastic photon detection for each view and regime.

Fixed-size preprocessing also uses base seed 1234, but derives stable sample-specific NumPy seeds from the MD5 hash of each relative file path. The clean branch and each corrupted branch use separate path-derived seeds before farthest-point or random sampling. Dataset splitting uses an independent seed of 38383 at the object-ID level, as detailed in Appendix B.1. These choices make benchmark generation and fixed-size sampling independent of multiprocessing order while keeping the reported split fixed.

B Validation and Characterization of the Single-Photon Benchmark

We use a set of diagnostic experiments to assess simulator consistency and characterize the resulting corruption. We first document benchmark assembly and object-level split integrity. We then verify ray-derived clean geometry and Poisson photon statistics, examine severity progression before and after fixed-size preprocessing, and compare single-photon corruption (SPC) with magnitude-matched isotropic Gaussian perturbations. These quantities characterize the benchmark and remain separate from the restoration metrics used in Sec. 5.

B.1 Benchmark Assembly and Split Integrity

SPORE contains objects from 55 ShapeNet categories. The fixed split is constructed at the object-ID level rather than at the individual noisy–clean pair level. Object IDs are sorted, shuffled with

Table 3: Single-photon simulation parameters for the Low, Mid, and High corruption regimes. Acquisition geometry and fixed sensor parameters are shared across all regimes, whereas photon level, background activity, dropout, and measurement uncertainty vary with corruption severity.

Parameter	Low	Mid	High
Acquisition geometry			
Acquisition mode	full-body	full-body	full-body
Number of viewpoints	10	10	10
Camera radius	2.2	2.2	2.2
Ray-grid resolution	96×96	96×96	96×96
Field of view	45°	45°	45°
Maximum simulated return range	3.2	3.2	3.2
Fixed photon and sensor parameters			
Laser wavelength	850 nm	850 nm	850 nm
Number of pulses	1	1	1
Detector efficiency η_{det}	0.25	0.25	0.25
Optical efficiency η_{opt}	0.60	0.60	0.60
Atmospheric efficiency η_{atm}	0.98	0.98	0.98
Mean reflectivity μ_ρ	0.55	0.55	0.55
Signal-rate scale κ_{sim}	10^{-6}	10^{-6}	10^{-6}
Maximum signal mean $\lambda_{\max}$	50	50	50
Regime-dependent corruption parameters			
Pulse energy E_{pulse}	5×10^{-10} J	1×10^{-10} J	3×10^{-11} J
Background Poisson mean λ_{bg}	0.02	0.15	0.45
Dark-count Poisson mean λ_{dark}	0.005	0.03	0.08
Base dropout probability p_0	0.03	0.12	0.25
Timing jitter σ_t	30 ps	80 ps	150 ps
Angular jitter σ_θ	0.01°	0.03°	0.08°
Additional range noise σ_{range}	0.0005 m	0.002 m	0.006 m
Spurious-return probability p_{spur}	0.005	0.04	0.10
Reflectivity std. σ_ρ	0.08	0.18	0.30

split seed 38383, and partitioned using an $80\%/10\%/10\%$ train/validation/test ratio. All available corruption variants of a given object are therefore assigned to the same split. Direct inspection of the split manifest confirms zero object-ID overlap among the training, validation, and test partitions.

The resulting split contains 41,880 training objects, 5,235 validation objects, and 5,235 test objects. Because fixed-size preprocessing excludes a regime-specific pair when the clean cloud or the radius-cropped corrupted cloud contains fewer than 2,048 points, the number of valid pairs differs slightly across corruption regimes. Table 4 reports the exact counts stored in the split manifest. In total, the benchmark contains 156,868 valid noisy–clean pairs.

B.2 Diagnostic Metrics

Let P denote a corrupted point cloud and G its clean reference. We compute nearest-neighbor distances in both directions, $d_{P\to G}(p) = \min_{g\in G} \|p-g\|_2, \qquad d_{G\to P}(g) = \min_{p\in P} \|g-p\|_2$. The diagnostic symmetric squared Chamfer distance is

$$\text{CD-L2}(P,G) = \frac{1}{|P|}\sum_{p\in P} d_{P\to G}(p)^2 + \frac{1}{|G|}\sum_{g\in G} d_{G\to P}(g)^2. \tag{20}$$

Diagnostic CD-L2 values in this appendix are reported directly in normalized-coordinate units.

The diagnostic F-score uses a Euclidean distance threshold of $\tau = 0.01$. Precision and recall are $\mathrm{Prec}_\tau = \frac{1}{|P|}\sum_{p\in P} \mathbb{I}\left[d_{P\to G}(p) \le \tau\right]$, $\mathrm{Rec}_\tau = \frac{1}{|G|}\sum_{g\in G} \mathbb{I}\left[d_{G\to P}(g) \le \tau\right]$, with $F@\tau = \frac{2\,\mathrm{Prec}_\tau\,\mathrm{Rec}_\tau}{\mathrm{Prec}_\tau+\mathrm{Rec}_\tau+10^{-12}}$. We report $F@0.01$. The outlier ratio measures the fraction of corrupted points whose nearest clean point is farther than 0.03, $R_{\mathrm{out}} = \frac{1}{|P|}\sum_{p\in P} \mathbb{I}\left[d_{P\to G}(p) > 0.03\right]$.

Table 4: Object-level split and valid noisy–clean pair counts in SPORE. All available corruption variants of the same object remain in one split.

Split	Objects	Low	Mid	High	Total pairs
Train	41,880	41,880	41,846	41,769	125,495
Validation	5,235	5,235	5,231	5,223	15,689
Test	5,235	5,235	5,230	5,219	15,684

To characterize local sampling structure, for each point $p_i \in P$ we compute the mean distance to its $K = 16$ nearest neighbors in the same point cloud, $s_i(P) = \frac{1}{K} \sum_{p_j \in \mathcal{N}_K(p_i;P)} \|p_i - p_j\|_2$. The reported mean nearest-neighbor distance is $|P|^{-1} \sum_i s_i(P)$. The quantity reported as local-density variance is the variance of these local mean spacings, $V_{\text{density}}(P) = \text{Var}_i [s_i(P)]$. Thus, larger values indicate stronger spatial variation in local sampling spacing; the statistic is used as a heterogeneity descriptor rather than as an inverse-density estimator.

Thin-structure recall is implemented as a sparse-structure proxy derived only from the clean reference. We compute $s_i(G)$ with the same $K = 16$ neighborhood and define the thin subset as the clean points in the upper 10% of local spacing, $G_{\text{thin}} = \left\{ g_i \in G : s_i(G) \geq Q_{0.90}\big(\{s_j(G)\}\big) \right\}$, where $Q_{0.90}$ denotes the 90th percentile. Thin-structure recall at 0.01 is then

$$R_{\text{thin}} = \frac{1}{|G_{\text{thin}}|} \sum_{g \in G_{\text{thin}}} \mathbb{I}\left[d_{G \to P}(g) \leq 0.01 \right]. \tag{21}$$

This operational definition targets locally sparse clean regions, which frequently include thin or weakly sampled structures.

For the raw-stage 500-model audit, point counts and detection ratios are computed from the complete raw clouds, whereas pairwise diagnostic metrics are evaluated on random subsets of at most 4,096 clean and corrupted points to control computational cost. The local-spacing calculation likewise uses at most 4,096 corrupted points. After fixed-size preprocessing, all 2,048 points are used for the corresponding diagnostics.

B.3 Geometric Validity of Ray-Derived Clean Targets

We evaluate the same ray–mesh intersection implementation used by the benchmark on four analytically interpretable toy geometries: a triangulated unit sphere, a planar square, a cube, and a thin cylindrical side surface. The simulator normalizes each toy mesh using the same bounding-box centering and maximum-radius rule as ShapeNet objects before tracing rays.

The analytic checks follow the geometry of each normalized primitive. Sphere error is $|\|\mathbf{x}\|_2 - 1|$; plane error is the absolute signed distance to $z = 0$; cube error is the minimum distance to the expected normalized face coordinate $1/\sqrt{3}$; and thin-cylinder error is the absolute difference between radial distance and the normalized cylinder radius. For the cube, surface consistency denotes the fraction of intersections with face error at most 10^{-4}.

As reported in Table 5, the plane and cube errors are close to numerical precision and all cube intersections satisfy the 10^{-4} face criterion. The sphere and thin cylinder have small analytic-surface deviations because the validation meshes are finite polygonal approximations of the corresponding continuous surfaces. These checks verify that the noiseless targets are generated from the intended visible ray–surface intersections.

B.4 Poisson Statistical Consistency

We next isolate the photon-count sampler from the remaining corruption mechanisms. The validation uses 1,000 independent trials with random seed 1234, range $d = 1.0$, normal incidence, fixed reflectivity $\rho = 0.55$, pulse energy 10^{-12} J, wavelength 850 nm, detector efficiency 0.25, optical efficiency 0.60, and atmospheric efficiency 0.98. Reflectivity variation, dropout, timing jitter, angular jitter, range noise, and spurious detections are disabled for this unit test. The background and dark-count rates are 0.02 and 0.01, giving $\lambda_b = 0.03$.

Table 5: Geometric validation of the ray-derived clean targets. Errors are measured against the analytic surface associated with each normalized toy geometry.

Geometry	GT points	Mean error	Max. error	Criterion
Sphere	8,880	0.001727	0.003387	Radius error
Plane	5,476	9.28×10^{-8}	4.77×10^{-7}	Signed plane distance
Cube	2,150	6.02×10^{-8}	2.38×10^{-7}	Surface consistency $= 1$
Thin cylinder	768	0.000126	0.000146	Radial surface error

Table 6 compares the configured Poisson quantities with empirical counts. For signal photons, the theoretical mean is 0.34596, while the empirical mean and variance are 0.33700 and 0.31543. For background photons, the theoretical mean is 0.03000, while the empirical mean and variance are 0.02900 and 0.03016. The theoretical probabilities of observing at least one photon are 0.29246 for signal and 0.02955 for background, compared with empirical probabilities of 0.29600 and 0.02800. The close agreement between theoretical and empirical statistics verifies the consistency of the implemented Poisson counting process under the controlled acquisition condition.

B.5 Severity Progression Before and After Fixed-Size Preprocessing

We evaluate Low, Mid, and High on 500 complete object triples spanning all 55 ShapeNet categories. The stored audit set contains 9–10 objects per category and no failed metric evaluations. We first analyze the raw detections before fixed-size preprocessing. Full-cloud point counts show that measurement availability decreases with corruption: the mean number of detected points decreases from $30{,}932.5$ under Low to $27{,}944.9$ under Mid and $24{,}118.2$ under High, while the detection ratio decreases from 0.9254 to 0.8346 and 0.7173.

The pairwise diagnostics show the same overall severity progression. Raw CD-L2 and outlier ratio increase, whereas F@0.01 and thin-structure recall decrease. After conversion to 2,048 points, the same ordering remains for the principal geometric diagnostics, indicating that the severity progression is already present in the raw acquisition statistics and persists after fixed-size preprocessing. Table 7 reports the mean and standard deviation across the 500 objects.

The per-object results provide an additional consistency check. At the raw stage, all 500 objects satisfy the expected Low-to-Mid-to-High ordering for detection ratio, CD-L2, F@0.01, outlier ratio, and mean nearest-neighbor distance; thin-structure recall follows the expected ordering for 497 of 500 objects (99.4%). After 2,048-point preprocessing, CD-L2, F@0.01, outlier ratio, and mean nearest-neighbor distance follow the expected ordering for all 500 objects, while thin-structure recall does so for 439 objects (87.8%). Local-density variance serves as a descriptor of spatial heterogeneity and exhibits a non-monotonic trend, reaching its maximum under Mid corruption at both stages. This behavior reflects the joint evolution of local spacing variation, measurement loss, and geometric perturbation across the corruption regimes.

B.6 Mean-Distance-Matched Gaussian Control

We construct an isotropic Gaussian control on the same test objects used for the SPC diagnostics. Importantly, the Gaussian scale is selected by matching a sampled-reference nearest-neighbor statistic rather than the Euclidean norm of the injected Gaussian displacement.

Let G_{raw} denote the raw clean point cloud and let P_{SPC} denote the corresponding raw SPC observation after the benchmark radius-2.0 crop. For scale selection, we define $\mathcal{S}_K(\cdot)$ as random sampling without replacement to at most $K = 4096$ points, leaving smaller point sets unchanged. A fixed sampled clean reference is first constructed as

$$\widetilde{G} = \mathcal{S}_K\left(G_{\mathrm{raw}}\right). \tag{22}$$

The corresponding SPC subset is

$$\widetilde{P}_{\mathrm{SPC}} = \mathcal{S}_K\left(P_{\mathrm{SPC}}\right). \tag{23}$$

Table 6: Monte Carlo validation of the photon-count sampler over 1,000 repeated trials under a fixed controlled acquisition condition.

Statistic	Signal	Background
Theoretical mean λ	0.34596	0.03000
Empirical count mean	0.33700	0.02900
Empirical count variance	0.31543	0.03016
Theoretical $P(N \geq 1)$	0.29246	0.02955
Empirical $P(N \geq 1)$	0.29600	0.02800

Table 7: Diagnostic corruption statistics across Low, Mid, and High regimes before and after fixed-size preprocessing, evaluated on 500 objects spanning 55 categories. Values are mean ± standard deviation. Raw point counts use complete clouds; raw pairwise diagnostics use up to 4,096 points per cloud. CD-L2 is reported in native normalized-coordinate units.

Diagnostic statistic	Low	Mid	High
Raw detections			
Number of points	$30{,}932.5 \pm 15{,}752.4$	$27{,}944.9 \pm 14{,}311.6$	$24{,}118.2 \pm 12{,}527.9$
Detection ratio	0.9254 ± 0.0033	0.8346 ± 0.0061	0.7173 ± 0.0118
CD-L2	0.00454 ± 0.00130	0.04285 ± 0.01063	0.19244 ± 0.05019
F@0.01	0.3278 ± 0.1975	0.2083 ± 0.1396	0.0927 ± 0.0662
Outlier ratio@0.03	0.0928 ± 0.0790	0.1732 ± 0.0813	0.4747 ± 0.0596
Mean NN distance	0.0519 ± 0.0123	0.0729 ± 0.0110	0.1029 ± 0.0101
Local-density variance	0.00320 ± 0.00108	0.00898 ± 0.00128	0.00557 ± 0.00097
Thin-structure recall@0.01	0.2908 ± 0.1914	0.1841 ± 0.1359	0.0809 ± 0.0649
2,048-point observations			
CD-L2	0.00393 ± 0.00141	0.03196 ± 0.00884	0.14408 ± 0.04149
F@0.01	0.1816 ± 0.1444	0.1135 ± 0.0965	0.0509 ± 0.0463
Outlier ratio@0.03	0.1098 ± 0.1218	0.1956 ± 0.1258	0.4927 ± 0.0915
Mean NN distance	0.06985 ± 0.01744	0.09403 ± 0.01524	0.13428 ± 0.01351
Local-density variance	0.00246 ± 0.00130	0.01301 ± 0.00240	0.00960 ± 0.00140
Thin-structure recall@0.01	0.1698 ± 0.1318	0.1077 ± 0.0895	0.0474 ± 0.0446

For a point set P, the matching statistic is the one-way mean nearest-neighbor distance to the sampled clean reference,

$$m(P, \widetilde{G}) = \frac{1}{|P|} \sum_{p \in P} \min_{g \in \widetilde{G}} \|p - g\|_2. \tag{24}$$

The SPC target for each object and corruption regime is therefore

$$m_{\mathrm{SPC}} = m\left(\widetilde{P}_{\mathrm{SPC}}, \widetilde{G}\right). \tag{25}$$

For a candidate Gaussian standard deviation σ, Gaussian perturbation is first applied to the raw clean coordinates,

$$P_{\mathrm{G}}^{\mathrm{raw}}(\sigma) = G_{\mathrm{raw}} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon}_i \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}\right). \tag{26}$$

The perturbed cloud is then radius-cropped and subsampled,

$$\widetilde{P}_{\mathrm{G}}(\sigma) = \mathcal{S}_K\left(\mathrm{Crop}_{2.0}\left(P_{\mathrm{G}}^{\mathrm{raw}}(\sigma)\right)\right). \tag{27}$$

The sampled clean reference $\widetilde{G}$ is held fixed during the candidate search for a given object and regime. The selected Gaussian scale is

$$\sigma^\star = \arg\min_{\sigma \in \mathcal{S}} \left| m\left(\widetilde{P}_{\mathrm{G}}(\sigma), \widetilde{G}\right) - m_{\mathrm{SPC}} \right|, \tag{28}$$

where $\mathcal{S}$ denotes the configured finite candidate set.

After $\sigma^\star$ is selected independently for each object and corruption regime, the Gaussian observation is regenerated from the raw clean cloud, radius-cropped, and randomly sampled without replacement to 2,048 points. Its clean reference is obtained from the same raw clean geometry using farthest-point sampling, following the benchmark evaluation pipeline.

Panel (a) of Table 8 reports the achieved matching quality. The average selected Gaussian standard deviations are 0.00244, 0.02621, and 0.09231 for Low, Mid, and High, respectively. The corresponding relative mismatch between the SPC target statistic and the Gaussian statistic is 2.7%, 0.6%, and 0.1%. Thus, the control approximately matches the sampled-reference one-way mean distance while retaining a different corruption process.

B.7 Spatial Degradation Characteristics of SPC and Gaussian Perturbation

SPC and the mean-distance-matched Gaussian control exhibit different spatial statistics despite similar sampled-reference one-way mean distances. The clearest separation occurs in local-spacing heterogeneity. Under Mid corruption, the reported local-density variance is 0.013440 for SPC and 0.000195 for Gaussian corruption; under High corruption, the corresponding values are 0.009947

Table 8: Comparison between single-photon corruption (SPC) and isotropic Gaussian perturbation on matched samples. Panel (a) reports per-model mean-distance matching. Panel (b) reports diagnostic statistics as SPC/Gaussian.

(a) Mean-distance matching

Level	Mean σ	Target mean dist.	Gaussian mean dist.	Rel. error
Low	0.00244	0.02158	0.02099	2.7%
Mid	0.02621	0.03147	0.03129	0.6%
High	0.09231	0.07226	0.07219	0.1%

(b) SPC versus matched Gaussian corruption

Level	CD-L2 (SPC / G)	F@0.01 (SPC / G)	Outlier@0.03 (SPC / G)	Density var. (SPC / G)	Thin recall@0.01 (SPC / G)
Low	0.00394 / 0.00109	0.199 / 0.192	0.081 / 0.083	0.002585 / 0.000156	0.187 / 0.178
Mid	0.03281 / 0.00196	0.124 / 0.049	0.167 / 0.370	0.013440 / 0.000195	0.118 / 0.040
High	0.14831 / 0.00924	0.0546 / 0.0154	0.472 / 0.759	0.009947 / 0.000679	0.0517 / 0.0123

and 0.000679. Based on the reported values, these correspond to approximately $68.9\times$ and $14.7\times$ larger local-density variance for SPC.

The diagnostic metrics reveal distinct aspects of the two corruption processes and consequently vary in different directions. Under Mid and High corruption, SPC yields CD-L2 values of 0.03281 and 0.14831, compared with 0.00196 and 0.00924 for the matched Gaussian control, corresponding to approximately $16.8\times$ and $16.1\times$ larger CD-L2. The Gaussian control, meanwhile, yields higher outlier ratios at Mid and High (0.370 and 0.759, compared with 0.167 and 0.472 for SPC) and lower thin-structure recall (0.040 and 0.0123, compared with 0.118 and 0.0517 for SPC). Together with the substantially larger local 16-NN spacing variance under SPC, these complementary statistics show that matched average displacement can correspond to markedly different spatial degradation patterns. In particular, SPC exhibits stronger spatial variation in local sampling support.

C Exact Evaluation Metric Definitions

We evaluate restoration using CD-L2, bidirectional P2M, F@0.01, and HD95. Let $\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^{N_P}$ denote the restored point cloud, $\mathbf{G} = \{\mathbf{g}_j\}_{j=1}^{N_G}$ the clean reference, and $\mathcal{M} = \{T_k\}_{k=1}^{F}$ the reference triangular mesh. All distances are Euclidean and are evaluated in the normalized SPORE coordinate system. For P2M, the reference OBJ is centered at its bounding-box midpoint and scaled by the maximum vertex radius, matching benchmark generation. The F-score threshold 0.01 therefore uses normalized coordinate units.

C.1 Chamfer Distance with Squared Euclidean Distance

For each restored point, we compute the distance to its nearest clean reference point, $d_{P\to G}(\mathbf{p}_i) = \min_{\mathbf{g}_j\in\mathbf{G}} \|\mathbf{p}_i - \mathbf{g}_j\|_2$, and analogously, $d_{G\to P}(\mathbf{g}_j) = \min_{\mathbf{p}_i\in\mathbf{P}} \|\mathbf{g}_j - \mathbf{p}_i\|_2$. CD-L2 is the sum of the two directional mean squared nearest-neighbor distances,

$$\text{CD-L2}(\mathbf{P},\mathbf{G}) = \frac{1}{N_P}\sum_{i=1}^{N_P} d_{P\to G}(\mathbf{p}_i)^2 + \frac{1}{N_G}\sum_{j=1}^{N_G} d_{G\to P}(\mathbf{g}_j)^2. \tag{29}$$

The $P \to G$ term measures the deviation of restored points from the clean reference, while the $G \to P$ term measures the coverage of the clean reference by the restored point set. In the quantitative tables, CD-L2 is reported in units of 10^{-3}, $\text{CD-L2}_{\text{table}} = 10^3\,\text{CD-L2}$.

C.2 Direct Bidirectional Point-to-Mesh Discrepancy

P2M evaluates the restored point cloud directly against the reference triangular mesh. For a point $\mathbf{p}$ and triangular face T, we define the point-to-triangle distance as $d(\mathbf{p},T) = \min_{\mathbf{y}\in T}\|\mathbf{p}-\mathbf{y}\|_2$. The prediction-to-mesh term is $D_{P\to M} = \frac{1}{N_P}\sum_{i=1}^{N_P}\min_{1\le k\le F} d(\mathbf{p}_i,T_k)^2$. For the reverse direction, each reference face contributes its minimum squared distance to any restored point, $D_{M\to P} = \frac{1}{F}\sum_{k=1}^{F}\min_{1\le i\le N_P} d(\mathbf{p}_i,T_k)^2$. The reverse term uses equal face weighting and exact point-to-triangle distances on the reference mesh. It does not resample the mesh surface into a

point cloud. The bidirectional metric is $\mathrm{P2M}(\mathbf{P}, \mathcal{M}) = D_{P\to M} + D_{M\to P}$. The first term measures prediction-to-surface discrepancy, whereas the second measures direct face-wise surface-to-prediction discrepancy. P2M is reported in units of 10^{-3}, so $\mathrm{P2M}_{\mathrm{table}} = 10^3\,\mathrm{P2M}$.

C.3 F-SCORE AT A DISTANCE THRESHOLD OF 0.01

Using the same directional nearest-neighbor distances as in CD-L2, precision at threshold τ is

$$\mathrm{Prec}_\tau = \frac{1}{N_P}\sum_{i=1}^{N_P} \mathbb{I}\left[d_{P\to G}(\mathbf{p}_i) \le \tau\right], \tag{30}$$

and recall is

$$\mathrm{Rec}_\tau = \frac{1}{N_G}\sum_{j=1}^{N_G} \mathbb{I}\left[d_{G\to P}(\mathbf{g}_j) \le \tau\right]. \tag{31}$$

The corresponding F-score is

$$F_\tau = \frac{2\,\mathrm{Prec}_\tau\,\mathrm{Rec}_\tau}{\mathrm{Prec}_\tau + \mathrm{Rec}_\tau + 10^{-12}}. \tag{32}$$

We set $\tau = 0.01$, $\quad \mathrm{F@0.01} = F_{0.01}$. The threshold is applied to the Euclidean nearest-neighbor distances. Precision measures point-wise accuracy of the restored set, while recall measures coverage of the clean reference. F@0.01 is dimensionless and is reported directly.

C.4 95TH-PERCENTILE BIDIRECTIONAL HAUSDORFF DISTANCE

We collect the two directional Euclidean-distance vectors $\mathbf{d}_{P\to G} = (d_{P\to G}(\mathbf{p}_i))_{i=1}^{N_P}$, and $\mathbf{d}_{G\to P} = (d_{G\to P}(\mathbf{g}_j))_{j=1}^{N_G}$. They are concatenated while retaining all entries,

$$\mathbf{d}_{\mathrm{bi}} = \mathrm{concat}(\mathbf{d}_{P\to G}, \mathbf{d}_{G\to P}), \tag{33}$$

and HD95 is defined as the empirical 95th percentile, $\mathrm{HD95}(\mathbf{P}, \mathbf{G}) = Q_{0.95}(\mathbf{d}_{\mathrm{bi}})$, where $Q_{0.95}$ denotes the empirical percentile operator. HD95 characterizes the upper tail of the pooled bidirectional geometric-error distribution. HD95 is computed from Euclidean distances and is reported directly.

C.5 EVALUATION PROTOCOL AND TEST-SET AGGREGATION

The fixed benchmark partition is shared across all compared methods, with a common category composition and identical object identities in the training, validation, and test splits. All learning-based methods are trained on the common training split. For each evaluated method, the checkpoint or method configuration used for final evaluation is selected according to performance on the common validation split. Final evaluation is performed on the fixed test split.

Within each corruption regime, all methods use the same evaluation set, including identical object identities, category composition, and valid noisy–clean pairs. All four metrics are computed independently for each restored object. Let m_s denote the metric value for the s-th valid test pair in a given corruption regime. For a test subset containing S valid pairs, the reported regime-level value is the arithmetic mean $\overline{m} = \frac{1}{S}\sum_{s=1}^{S} m_s$. The fixed test partition contains 5,235 object IDs. After the regime-specific validity filtering used during fixed-size benchmark construction, the Low, Mid, and High evaluation sets contain 5,235, 5,230, and 5,219 valid noisy–clean pairs, respectively. For each regime, the reported value is the arithmetic mean over this shared evaluation set.

D DETAILED ARCHITECTURE AND GALR IMPLEMENTATION

This section provides the implementation details required to reproduce Geometry-Aware Latent Repair (GALR) and repair-state-conditioned Global-Local Flow Matching (GLFM). We first define the latent interface and standardization, then specify reliability estimation, anchor repair, repair-state construction, the two velocity networks, clean-local alignment, and latent-state evolution. Table 9 summarizes the principal architectural settings.

D.1 Latent Representation and Standardization

For an input point cloud $\mathbf{X}$, the frozen hierarchical encoder produces $(\mathbf{Z}_g, \mathbf{Z}_\ell) = E(\mathbf{X}), \qquad \mathbf{Z}_\ell = \{\mathbf{z}_{\ell,i}\}_{i=1}^{M}$. Here, $\mathbf{Z}_g$ denotes the object-level latent vector, while $\mathbf{Z}_\ell$ denotes the set of spatially organized local latent tokens.

The global latent dimension is $d_g = 128$, and each local token has dimension $d_\ell = 4$, $\mathbf{z}_{\ell,i} = [\mathbf{x}_i, \mathbf{f}_i], \qquad \mathbf{x}_i \in \mathbb{R}^3, \quad \mathbf{f}_i \in \mathbb{R}$. The local encoder preserves one spatially aligned local token per input point. With the 2,048-point benchmark representation, this gives $M = 2048$ local tokens.

Flow Matching operates on standardized latent variables. Training-set statistics are computed separately for global and local channels by pooling both clean and noisy encoder endpoints. For a scalar channel ξ of either latent representation, let ξ^{n} and ξ^{c} denote its noisy and clean endpoint values. The stored endpoint statistics are

$$\mu_{\mathrm{end}} = \mathrm{Mean}(\xi^{\mathrm{n}}, \xi^{\mathrm{c}}), \qquad \sigma_{\mathrm{end}} = \sqrt{\mathrm{Var}(\xi^{\mathrm{n}}, \xi^{\mathrm{c}}) + 10^{-6}}, \tag{34}$$

and the standardized channel is $\bar{\xi} = \frac{\xi - \mu_{\mathrm{end}}}{\sigma_{\mathrm{end}}}$. All Flow Matching states and conditions below use this standardized latent representation unless stated otherwise. GALR converts the first three local channels back to the encoder anchor coordinates when computing spatial distances. We denote the resulting de-standardized anchor by $\mathbf{x}_i^{\mathrm{anc}}$, $\mathbf{x}_i^{\mathrm{anc}} = \bar{\mathbf{x}}_i \odot \boldsymbol{\sigma}_{\ell,xyz} + \boldsymbol{\mu}_{\ell,xyz}$. Restored endpoint latents are transformed back to the decoder latent coordinates before reconstruction.

GALR updates the spatial component of each local token, $\widetilde{\mathbf{z}}_{\ell,i}^{\mathrm{n}} = [\widetilde{\mathbf{x}}_i^{\mathrm{n}}, \mathbf{f}_i^{\mathrm{n}}]$, thereby preserving the associated latent feature during pre-repair.

D.2 Geometry-Aware Latent Repair

Geometry prior and local structural descriptors. For each noisy anchor, the geometric deviation statistic uses $K_{\mathrm{dev}} = 32$ nearest neighbors, with the query point excluded from the pre-repair KNN search. Let $\bar{d}_i^{\mathrm{n}}$ denote the mean Euclidean distance over these neighbors. Within each observation, the standardized deviation is $s_i^{\mathrm{n}} = \frac{\bar{d}_i^{\mathrm{n}} - \mu_d^{\mathrm{n}}}{\max(\sigma_d^{\mathrm{n}}, 10^{-6})}$. The geometry-derived corruption prior is $\pi_i^{\mathrm{geo}} = \sigma\left(\frac{s_i^{\mathrm{n}} - \alpha}{\tau}\right), \qquad \alpha = 2.0, \quad \tau = 0.25.$

Structural descriptors are computed from $K_{\mathrm{str}} = 16$ neighbors. Let $\lambda_{i,1} \geq \lambda_{i,2} \geq \lambda_{i,3} \geq 0$ be the eigenvalues of the local coordinate covariance matrix, and define $\bar{\lambda}_{i,1} = \max(\lambda_{i,1}, 10^{-8})$. The implementation computes

$$\begin{aligned} L_i &= \mathrm{clip}\left(\frac{\lambda_{i,1} - \lambda_{i,2}}{\bar{\lambda}_{i,1}}, 0, 1\right), \\ P_i &= \mathrm{clip}\left(\frac{\lambda_{i,2} - \lambda_{i,3}}{\bar{\lambda}_{i,1}}, 0, 1\right), \\ S_i &= \mathrm{clip}\left(\frac{\lambda_{i,3}}{\bar{\lambda}_{i,1}}, 0, 1\right), \\ G_i &= \max(L_i, P_i)(1 - S_i). \end{aligned} \tag{35}$$

The resulting local geometry descriptor is $\mathbf{g}_i^{\mathrm{n}} = [s_i^{\mathrm{n}}, \pi_i^{\mathrm{geo}}, L_i, P_i, S_i, G_i]$.

Object-level corruption statistics and reliability calibration. The object-level corruption statistics summarize the distribution of geometry-derived priors within the current observation,

$$\boldsymbol{\nu} = [\mathrm{Mean}_i(\pi_i^{\mathrm{geo}}), \mathrm{Std}_i(\pi_i^{\mathrm{geo}}), \mathrm{Top10Mean}_i(\pi_i^{\mathrm{geo}}), \mathrm{Max}_i(\pi_i^{\mathrm{geo}})]. \tag{36}$$

Here, $\mathrm{Top10Mean}$ averages the largest 10% of point-wise priors. The statistic is embedded by a $4 \to 16 \to 8$ MLP with SiLU activation. The token-wise calibration input concatenates the 4-D local latent, 128-D global latent, 8-D object-level statistic embedding, and 6-D geometry descriptor. A two-hidden-layer point network therefore maps $146 \to 128 \to 128$ with SiLU activation after each hidden layer.

Let $\mathbf{h}_i^{\mathrm{cal}} \in \mathbb{R}^{128}$ denote the resulting calibration feature. Two scalar heads refine the prior logit,

$$\Delta_i^{\mathrm{base}} = 2\tanh\big(h_{\mathrm{base}}(\mathbf{h}_i^{\mathrm{cal}})\big),$$
$$\Delta_i^{\mathrm{extra}} = 2\tanh\left(\frac{h_{\mathrm{extra}}(\mathbf{h}_i^{\mathrm{cal}})}{2}\right), \tag{37}$$

and the calibrated corruption probability and reliability are

$$p_i = \sigma\big(\mathrm{logit}(\pi_i^{\mathrm{geo}}) + \Delta_i^{\mathrm{base}} + \Delta_i^{\mathrm{extra}}\big), \qquad r_i = 1 - p_i. \tag{38}$$

Both scalar heads are initialized at zero, so calibration begins from the geometry-derived prior.

D.3 Reliability-Based Anchor Repair

Repair-neighborhood construction. For each anchor, GALR first forms a candidate pool containing its $K_{\mathrm{cand}} = 64$ nearest spatial neighbors. A candidate is treated as reliable when $p_j < \tau_{\mathrm{rel}}, \qquad \tau_{\mathrm{rel}} = 0.5$. The repair neighborhood $\mathcal{R}_i$ contains up to $K_{\mathrm{rep}} = 8$ closest reliable candidates. If between one and eight reliable candidates are available, interpolation is normalized over the available reliable subset. If no reliable candidate is available, the implementation uses the mean anchor coordinate of the eight nearest unfiltered candidates as the geometric fallback.

For a valid pair (i, j), the relation descriptor is $\mathbf{e}_{ij} = [\Delta\mathbf{x}_{ij}, \log(\max(d_{ij}, 10^{-6})), \Delta f_{ij}, p_j, s_j^{\mathrm{n}}, G_j]$, where $\Delta\mathbf{x}_{ij}$ is evaluated in de-standardized anchor coordinates. With one local latent feature, $\mathbf{e}_{ij} \in \mathbb{R}^8$. A relation MLP $8 \to 32 \to 1$ uses SiLU activation after the hidden layer. Its score is combined with inverse-distance weighting, $a_{ij} = -\log(\max(d_{ij}, 10^{-6})) + 2\tanh(h_\omega(\mathbf{e}_{ij}))$, and normalized across the valid repair neighbors, $\omega_{ij} = \frac{\exp(a_{ij})}{\sum_{k\in\mathcal{R}_i}\exp(a_{ik})}$. The geometry-supported anchor and displacement are

$$\mathbf{x}_i^{\mathrm{geo}} = \sum_{j\in\mathcal{R}_i} \omega_{ij}\mathbf{x}_j^{\mathrm{n}}, \qquad \Delta\mathbf{x}_i^{\mathrm{geo}} = \mathbf{x}_i^{\mathrm{geo}} - \mathbf{x}_i^{\mathrm{n}}. \tag{39}$$

The weighted 32-D hidden relation features are pooled with the same ω_{ij} to form a neighborhood context $\mathbf{h}_i^{\mathrm{nbr}} \in \mathbb{R}^{32}$.

Bounded residual refinement. The residual repair network receives $[\mathbf{h}_i^{\mathrm{cal}}, \Delta\mathbf{x}_i^{\mathrm{geo}}, p_i, \mathbf{h}_i^{\mathrm{nbr}}] \in \mathbb{R}^{164}$ and applies $164 \to 128 \to 3$ with SiLU activation after the hidden layer, producing $\Delta\mathbf{x}_i^{\mathrm{raw}}$. Let $\|\cdot\|_{\mathrm{geo}}$ denote the Euclidean norm after applying the local-coordinate standard deviation. The implementation uses

$$\gamma_i = \min\left(1, 0.20\,\frac{\|\Delta\mathbf{x}_i^{\mathrm{geo}}\|_{\mathrm{geo}}}{\max(\|\Delta\mathbf{x}_i^{\mathrm{raw}}\|_{\mathrm{geo}}, 10^{-8})}\right), \tag{40}$$

$$\Delta\mathbf{x}_i^{\mathrm{res}} = \gamma_i\Delta\mathbf{x}_i^{\mathrm{raw}}, \qquad \Delta\mathbf{x}_i^{\mathrm{rep}} = \Delta\mathbf{x}_i^{\mathrm{geo}} + \Delta\mathbf{x}_i^{\mathrm{res}}. \tag{41}$$

Thus, the residual component is bounded to at most 20% of the geometry-supported displacement in the de-standardized anchor metric.

Continuous repair and repair state. The calibrated corruption probability is mapped to a continuous repair weight, $w_i = \mathrm{clip}\left(\frac{p_i - 0.25}{0.75 - 0.25}, 0, 1\right)$. The pre-repaired anchor and realized correction are

$$\widetilde{\mathbf{x}}_i^{\mathrm{n}} = \mathbf{x}_i^{\mathrm{n}} + w_i\Delta\mathbf{x}_i^{\mathrm{rep}}, \qquad \boldsymbol{\delta}_i = \widetilde{\mathbf{x}}_i^{\mathrm{n}} - \mathbf{x}_i^{\mathrm{n}}. \tag{42}$$

The bounded repair magnitude is computed in standardized local-latent coordinates, $m_i = \tanh(\|\boldsymbol{\delta}_i\|_2)$, and the seven-dimensional repair state is $\mathbf{c}_i = [r_i, w_i, \pi_i^{\mathrm{geo}}, m_i, \tanh(\boldsymbol{\delta}_i)]$. The resulting local token is $\widetilde{\mathbf{z}}_{\ell,i}^{\mathrm{n}} = [\widetilde{\mathbf{x}}_i^{\mathrm{n}}, \mathbf{f}_i^{\mathrm{n}}]$.

D.4 Object-Level Repair Context

The object-level repair context is defined as

$$\mathbf{c}_g = \big[\mathrm{Mean}_i(\widetilde{\mathbf{z}}_{\ell,i}^{\mathrm{n}}), \mathrm{Mean}_i(\pi_i^{\mathrm{geo}}), \mathrm{Mean}_i(w_i), \mathrm{Mean}_i(m_i)\big], \tag{43}$$

giving $d_{c_g} = 4 + 1 + 1 + 1 = 7$. The object-level context is supplied to both the global and local velocity networks throughout latent transport. The token-wise repair-state set $\mathbf{C} = \{\mathbf{c}_i\}_{i=1}^M$ is additionally supplied to the local branch.

D.5 Global Velocity Network

The global branch models the evolution of the object-level global latent $\mathbf{Z}_t^g \in \mathbb{R}^{128}$. Continuous time $t \in [0, 1]$ is multiplied by 1000 before a 128-D sinusoidal embedding, followed by a two-layer time MLP $128 \to 128 \to 128$ with SiLU activation. The global velocity network receives the current global state, time embedding, and object-level repair context, $\mathbf{q}_t^g = [\mathbf{Z}_t^g, \mathbf{e}_t, \mathbf{c}_g] \in \mathbb{R}^{263}$. A four-layer MLP with hidden width 512 and SiLU activations maps this condition to a 128-D velocity, $\mathbf{v}_\theta^g = \mathbf{v}_\theta^g(\mathbf{Z}_t^g, t \mid \mathbf{c}_g)$. Dropout is set to zero.

D.6 Local KNN Transformer Velocity Network

The local velocity field is implemented with a four-block KNN Transformer. The network operates on the evolving local latent state $\mathbf{Z}_t^\ell$ and combines dynamic geometric neighborhoods with object-level and token-wise repair conditioning. Its detailed architecture is illustrated in Fig. 5.

Token embedding and neighborhood construction. Each four-dimensional local state is projected with $\mathrm{Linear}(4, 512)$, while its 3-D anchor is independently projected with $\mathrm{Linear}(3, 512)$; the two embeddings are added. The local velocity network contains four KNN Transformer blocks with hidden dimension 512, eight attention heads, head dimension 64, and FFN dimension 2,048. GELU is used in the FFN and dropout is zero. The local branch uses KNN Transformer blocks because local restoration requires dynamically updated geometric interactions among neighboring tokens during latent transport.

For the evolving local anchor $\mathbf{x}_i(t)$, the adjusted KNN distance is $\widetilde{d}_{ij}(t) = \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|_2 + 0.25(1-t)(1-r_j)$. Here, $\mathbf{x}_i(t)$ and $\mathbf{x}_j(t)$ denote the current spatial anchors of the center token i and candidate neighbor j, respectively, while r_j is the reliability of neighbor j. The coefficient 0.25 controls the reliability penalty. The $K = 16$ smallest adjusted distances define the neighborhood for each velocity evaluation. The implementation recomputes this graph from the current anchors whenever the velocity network is evaluated. The query token remains eligible during KNN selection, so a selected neighborhood may include the self token.

Geometric and reliability-biased attention. For neighbor j of token i, the relative descriptor $[\mathbf{x}_j(t) - \mathbf{x}_i(t), \|\mathbf{x}_j(t) - \mathbf{x}_i(t)\|_2]$ is mapped through $4 \to 512 \to 8$ with SiLU activation to produce one geometric bias per attention head. For head h,

$$\ell_{ij}^{(h)} = \frac{\mathbf{q}_i^{(h)\top}\mathbf{k}_j^{(h)}}{\sqrt{64}} + b_{\mathrm{geo},ij}^{(h)} + \log(\max(r_j, 10^{-4})). \tag{44}$$

Softmax normalization is applied over the 16 selected entries. Thus, the adjusted KNN distance controls where information is aggregated, whereas the reliability bias controls how strongly each selected neighbor contributes.

D.7 Object-Level AdaLN and Point-Wise FiLM

The local branch is conditioned on the evolving global state together with time and the object-level repair context, $\mathbf{q}_t^\ell = [\mathbf{Z}_t^g, \mathbf{e}_t, \mathbf{c}_g] \in \mathbb{R}^{263}$. This vector is embedded by $263 \to 512 \to 512$ with SiLU activation. For each Transformer block s, the object-level conditioner is mapped by $\mathrm{SiLU} \to \mathrm{Linear}(512, 6 \times 512)$ to six 512-D vectors,

$$[\boldsymbol{\beta}_{g,\mathrm{att}}^{(s)}, \boldsymbol{\gamma}_{g,\mathrm{att}}^{(s)}, \mathbf{g}_{\mathrm{att}}^{(s)}, \boldsymbol{\beta}_{g,\mathrm{ffn}}^{(s)}, \boldsymbol{\gamma}_{g,\mathrm{ffn}}^{(s)}, \mathbf{g}_{\mathrm{ffn}}^{(s)}]. \tag{45}$$

The two $\mathbf{g}$ vectors gate the attention and FFN residual branches channel-wise.

The seven-dimensional token-wise repair state is encoded by $7 \to 64$ with SiLU activation. Each block then applies an additional SiLU followed by $\mathrm{Linear}(64, 4 \times 512)$ to produce

$$[\boldsymbol{\beta}_{i,\mathrm{att}}^{p,(s)}, \boldsymbol{\gamma}_{i,\mathrm{att}}^{p,(s)}, \boldsymbol{\beta}_{i,\mathrm{ffn}}^{p,(s)}, \boldsymbol{\gamma}_{i,\mathrm{ffn}}^{p,(s)}]. \tag{46}$$

For a normalized feature $\mathbf{u}_i$, define the modulation operator

$$\mathrm{Mod}_i^{(s)}(\mathbf{u}_i; \boldsymbol{\beta}_g, \boldsymbol{\gamma}_g, \boldsymbol{\beta}_i^p, \boldsymbol{\gamma}_i^p) = (1 + \boldsymbol{\gamma}_g + \boldsymbol{\gamma}_i^p) \odot \mathbf{u}_i + \boldsymbol{\beta}_g + \boldsymbol{\beta}_i^p. \tag{47}$$

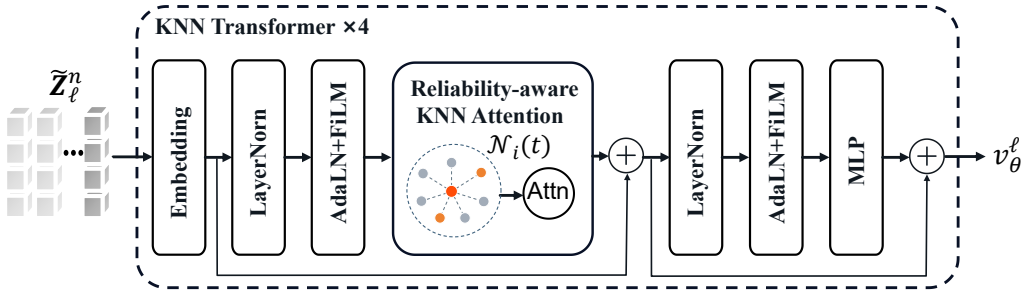


Figure 5: Architecture of the local KNN Transformer velocity network. The evolving local latent state $\mathbf{Z}_t^\ell$ is first embedded and then processed by four KNN Transformer blocks. Each block contains reliability-aware KNN attention, feature modulation, a feed-forward network, and residual connections. Object-level conditioning derived from $[\mathbf{Z}_t^g, \mathbf{e}_t, \mathbf{c}_g]$ is injected through AdaLN, while token-wise repair states modulate individual tokens through point-wise FiLM. Token reliability further affects neighborhood selection and attention weighting. The network outputs the local velocity field $\mathbf{v}_\theta^\ell$.

The attention and FFN sublayers are therefore

$$\begin{aligned}\mathbf{h}' &= \mathbf{h} + \mathbf{g}_{\text{att}}^{(s)} \odot \text{Attn}\Big(\text{Mod}_i^{(s)}(\text{LN}(\mathbf{h}))\Big), \\ \mathbf{h}'' &= \mathbf{h}' + \mathbf{g}_{\text{ffn}}^{(s)} \odot \text{FFN}\Big(\text{Mod}_i^{(s)}(\text{LN}(\mathbf{h}'))\Big).\end{aligned} \tag{48}$$

Layer normalization uses $\epsilon_{\text{LN}} = 10^{-6}$. The AdaLN, point-wise FiLM, final AdaLN, and final velocity projection are initialized with zero output weights and biases.

After four blocks, a final LayerNorm and object-conditioned affine modulation are followed by a $512 \to 4$ projection. The implementation bounds the four local velocity channels element-wise as $\mathbf{v}_\theta^\ell = 2\tanh\left(\frac{\mathbf{v}_{\text{raw}}^\ell}{2}\right)$. The output channels correspond to the three spatial-anchor velocities and one latent-feature velocity.

D.8 Clean-Local Latent Alignment

During training, the clean point cloud is encoded as $(\mathbf{Z}_g^{\text{c}}, \mathbf{Z}_\ell^{\text{c}}) = E(\mathbf{X}^{\text{c}})$. The local Flow Matching target is aligned to the pre-repaired initialization in standardized local-anchor coordinates. For each pre-repaired anchor,

$$a(i) = \arg\min_j \left\|\widetilde{\mathbf{x}}_i^{\text{n}} - \mathbf{x}_j^{\text{c}}\right\|_2, \qquad \mathbf{z}_{\ell,i}^{\text{tar}} = \mathbf{z}_{\ell,a(i)}^{\text{c}}. \tag{49}$$

The nearest-anchor search processes 512 query tokens per chunk and gathers the complete 4-D clean local latent after matching.

D.9 Condition and State Evolution During Latent Transport

GALR is evaluated once for each noisy input, producing $\widetilde{\mathbf{Z}}_\ell^{\text{n}}, \quad \mathbf{C}, \quad \mathbf{c}_g$. The repair conditions $\mathbf{C}$ and $\mathbf{c}_g$ remain fixed for the input, while $\mathbf{Z}_t^g$ and $\mathbf{Z}_t^\ell$ evolve during latent transport. The global path starts from $\mathbf{Z}_g^{\text{n}}$, and the local path starts from the pre-repaired state $\widetilde{\mathbf{Z}}_\ell^{\text{n}}$.

During training, the global branch follows the linear Flow Matching path

$$\mathbf{Z}_t^g = (1-t)\mathbf{Z}_g^{\text{n}} + t\mathbf{Z}_g^{\text{c}}, \qquad \mathbf{U}_g = \mathbf{Z}_g^{\text{c}} - \mathbf{Z}_g^{\text{n}}. \tag{50}$$

Using the same sampled time t, the local branch follows

$$\mathbf{Z}_t^\ell = (1-t)\widetilde{\mathbf{Z}}_\ell^{\text{n}} + t\mathbf{Z}_\ell^{\text{tar}}, \qquad \mathbf{U}_\ell = \mathbf{Z}_\ell^{\text{tar}} - \widetilde{\mathbf{Z}}_\ell^{\text{n}}. \tag{51}$$

The velocity networks are trained to predict $\mathbf{U}_g$ and $\mathbf{U}_\ell$ under the conditioning defined above.

At inference, the coupled latent dynamics use the same time-indexed notation as in the main text,

$$\begin{aligned} \frac{\mathrm{d}\mathbf{Z}_t^g}{\mathrm{d}t} &= \mathbf{v}_\theta^g\left(\mathbf{Z}_t^g, t \mid \mathbf{c}_g\right), & \mathbf{Z}_0^g &= \mathbf{Z}_g^{\mathrm{n}}, \\ \frac{\mathrm{d}\mathbf{Z}_t^\ell}{\mathrm{d}t} &= \mathbf{v}_\theta^\ell\left(\mathbf{Z}_t^\ell, t \mid \mathbf{Z}_t^g, \mathbf{c}_g, \mathbf{C}\right), & \mathbf{Z}_0^\ell &= \widetilde{\mathbf{Z}}_\ell^{\mathrm{n}}. \end{aligned} \tag{52}$$

The evolving global latent therefore provides object-level structural conditioning to the local velocity field, while the fixed repair contexts preserve the GALR-derived restoration information throughout transport. The local KNN graph is recomputed from the current spatial anchors at each velocity evaluation. Inference uses four Euler steps over $t \in [0,1]$. At the endpoint, $\widehat{\mathbf{Z}}_g = \mathbf{Z}_1^g, \quad \widehat{\mathbf{Z}}_\ell = \mathbf{Z}_1^\ell$, and the restored latents are transformed to decoder coordinates and reconstructed as $\mathbf{X}_{\mathrm{p}} = D\left(\widehat{\mathbf{Z}}_g, \widehat{\mathbf{Z}}_\ell\right)$.

E Complete Learning Objectives and Optimization Details

This section specifies the complete objectives and optimization protocol for GALR and the subsequent GLFM stage. Training follows the two-stage procedure in Sec. 4.4. We first optimize GALR with the hierarchical autoencoder fixed. GALR is then frozen while the global and local velocity fields are trained from fixed pre-repaired latents and repair conditions.

E.1 GALR Objective

GALR is trained with two complementary objectives. A reliability-calibration loss supervises the predicted corruption probability, while a structure-weighted bidirectional set loss constrains the geometry of the pre-repaired local anchors.

Anchor-space convention. Flow Matching operates on standardized latent variables, as defined in Appendix D.1. In contrast, the geometric distances used in the GALR supervision below are evaluated in the de-standardized encoder-anchor coordinate system used by GALR for geometry-aware operations. To distinguish these anchor sets from the input point clouds $\mathbf{X}^{\mathrm{n}}$ and $\mathbf{X}^{\mathrm{c}}$, we write

$$\mathcal{A}_\ell^{\mathrm{n}} = \{\mathbf{x}_i^{\mathrm{n}}\}_{i=1}^M, \qquad \mathcal{A}_\ell^{\mathrm{c}} = \{\mathbf{x}_j^{\mathrm{c}}\}_{j=1}^M, \tag{53}$$

where $\mathbf{x}_i^{\mathrm{n}}$ and $\mathbf{x}_j^{\mathrm{c}}$ denote de-standardized noisy and clean encoder-anchor coordinates, respectively.

Clean-aware calibration target. The clean latent geometry is used only to construct supervision during GALR training and is not available to the model at inference. At test time, reliability prediction depends only on the corrupted observation and the GALR inputs defined in Sec. 4.2.

For each noisy anchor, we compute its nearest-anchor distance to the paired clean latent geometry, $d_i^{\mathrm{nc}} = \min_j \left\|\mathbf{x}_i^{\mathrm{n}} - \mathbf{x}_j^{\mathrm{c}}\right\|_2$. Within each observation, these distances are standardized as

$$q_i = \frac{d_i^{\mathrm{nc}} - \mu_{\mathrm{nc}}}{\sigma_{\mathrm{nc}} + \epsilon}, \qquad \mu_{\mathrm{nc}} = \mathrm{Mean}_i(d_i^{\mathrm{nc}}), \qquad \sigma_{\mathrm{nc}} = \mathrm{Std}_i(d_i^{\mathrm{nc}}), \tag{54}$$

with $\epsilon = 10^{-6}$. The resulting soft corruption target is $y_i = \sigma\left(\frac{q_i - \alpha}{\tau_{\mathrm{tar}}}\right)$, where $\alpha = 2.0$ and $\tau_{\mathrm{tar}} = 0.25$. This target provides continuous training supervision for the calibrated corruption probability p_i defined in Sec. 4.2.

Focal reliability calibration. Let a_i denote the calibrated corruption logit, $p_i = \sigma(a_i), \qquad r_i = 1 - p_i$. For the soft target y_i, the binary cross-entropy term is $\ell_i^{\mathrm{BCE}} = -y_i \log p_i - (1-y_i)\log(1-p_i)$. We further define $p_{t,i} = y_i p_i + (1-y_i)(1-p_i)$, and the soft class-balancing factor $\alpha_i^{\mathrm{f}} = \alpha_{\mathrm{f}} y_i + (1-\alpha_{\mathrm{f}})(1-y_i)$. The focal calibration objective is

$$\mathcal{L}_{\mathrm{cal}} = \frac{1}{M}\sum_{i=1}^M \alpha_i^{\mathrm{f}} (1 - p_{t,i})^{\gamma_{\mathrm{f}}} \ell_i^{\mathrm{BCE}}, \tag{55}$$

where $\alpha_{\mathrm{f}} = 0.75$ and $\gamma_{\mathrm{f}} = 2.0$.

Table 9: Architecture and implementation settings for Geometry-Aware Latent Repair and the repair-state-conditioned Global-Local Flow Matching networks.

Component	Configuration
Global latent dimension	128
Local latent dimension	$4 = 3$D anchor $+1$ feature
Local latent tokens	2,048; one token per input point
Latent standardization	train-set pooled clean/noisy endpoint mean and std
Pre-repair deviation KNN	$K_{\mathrm{dev}} = 32$; self excluded
Structural KNN	$K_{\mathrm{str}} = 16$
Geometry prior	$\alpha = 2.0$, $\tau = 0.25$
Object-level statistics encoder	$4 \to 16 \to 8$, SiLU
Calibration point MLP	$146 \to 128 \to 128$, SiLU
Calibration heads	two $128 \to 1$ scalar heads
Repair candidate pool	64
Reliable-neighbor threshold	$p_j < 0.5$
Repair neighbors	up to 8; unfiltered 8-NN mean fallback if none reliable
Relation MLP	$8 \to 32 \to 1$, SiLU
Residual repair MLP	$164 \to 128 \to 3$, SiLU
Residual ratio	0.20 of geometry-supported metric displacement
Soft repair thresholds	0.25, 0.75
Repair-state dimension	7
Object-level repair context dimension	7
Time embedding	t multiplied by 1000, then 128-D sinusoidal + $128 \to 128 \to 128$ MLP
Global velocity MLP	input $[\mathbf{Z}_t^g, \mathbf{e}_t, \mathbf{c}_g]$; four 512-D hidden layers + 128-D output
Local velocity network	4 KNN Transformer blocks conditioned on $[\mathbf{Z}_t^g, \mathbf{e}_t, \mathbf{c}_g]$ and token-wise $\mathbf{C}$
Transformer hidden / heads / FFN	512 / 8 / 2,048
Local KNN	$K = 16$, recomputed at each velocity evaluation
Reliability KNN penalty	$0.25(1-t)(1-r_j)$
Attention reliability bias	$\log(\max(r_j, 10^{-4}))$
Point-wise FiLM encoder	$7 \to 64$, SiLU
Relative-position bias MLP	$4 \to 512 \to 8$, SiLU
Local velocity bound	$2\tanh(\mathbf{v}_{\mathrm{raw}}^{\ell}/2)$
Clean-local alignment	nearest clean anchor; chunk size 512
Inference solver	4-step Euler
Dropout	0

Structure-weighted bidirectional set loss. Let $\widetilde{\mathcal{A}}_\ell^{\mathrm{n}} = \{\widetilde{\mathbf{x}}_i^{\mathrm{n}}\}_{i=1}^{M}$ denote the de-standardized spatial anchors after continuous GALR correction. During training, 512 anchors are sampled from the repaired and clean anchor sets to evaluate the set-level objective. The repaired-to-clean direction is

$$\mathcal{L}_{\mathrm{p2c}} = \frac{1}{N_p} \sum_{i=1}^{N_p} \min_j \left\| \widetilde{\mathbf{x}}_i^{\mathrm{n}} - \mathbf{x}_j^{\mathrm{c}} \right\|_2^2 . \tag{56}$$

The reverse direction places additional training weight on clean anchors with sparse or structurally elongated local support. This weighting is used only as a training-time coverage prior and is distinct from the evaluation metrics in Appendix C. For each sampled clean anchor, we compute a local structural score from its $K_{\text{str-loss}} = 16$ nearest neighbors, $h_j^{\mathrm{str}} = Z_{\mathrm{r}}(s_j^{\mathrm{sp}}) + Z_{\mathrm{r}}(\kappa_j) + Z_{\mathrm{r}}(L_j)$, where $Z_{\mathrm{r}}(v) = \frac{v - \mathrm{median}(v)}{1.4826\ \mathrm{MAD}(v) + \epsilon}$. Here, s_j^{sp} is the mean neighbor spacing, κ_j is the smallest covariance eigenvalue divided by the eigenvalue sum, and L_j is local linearity. We mark the highest 15% of structural scores with $I_j^{\mathrm{str}} \in \{0, 1\}$ and define $\eta_j = 1 + \lambda_{\mathrm{str}} I_j^{\mathrm{str}}$, $\lambda_{\mathrm{str}} = 2.0$. The clean-to-repaired direction is

$$\mathcal{L}_{\mathrm{c2p}} = \frac{\sum_{j=1}^{N_c} \eta_j \min_i \left\| \mathbf{x}_j^{\mathrm{c}} - \widetilde{\mathbf{x}}_i^{\mathrm{n}} \right\|_2^2}{\sum_{j=1}^{N_c} \eta_j}. \tag{57}$$

The bidirectional repair loss is $\mathcal{L}_{\text{set}} = \mathcal{L}_{\text{p2c}} + \mathcal{L}_{\text{c2p}}$. Combining geometric repair and reliability calibration gives the GALR objective used in Sec. 4.4:

$$\boxed{\mathcal{L}_{\text{GALR}} = \mathcal{L}_{\text{set}} + \lambda_{\text{cal}}\mathcal{L}_{\text{cal}}}, \qquad \lambda_{\text{cal}} = 0.2. \tag{58}$$

The calibration term supervises which local anchors require stronger correction. The bidirectional set term constrains the geometry produced by that correction.

E.2 Global-Local Flow Matching Objective

After GALR optimization, the encoder E, decoder D, and GALR module are frozen. GLFM is then trained on paired noisy and clean hierarchical latents. All interpolation paths and velocity-regression losses in this stage are evaluated in the standardized latent space defined in Appendix D.1.

Global velocity regression. For $t \sim \mathcal{U}(0,1)$, the global interpolation path and target velocity are

$$\mathbf{Z}_t^g = (1-t)\mathbf{Z}_g^{\text{n}} + t\mathbf{Z}_g^{\text{c}}, \qquad \mathbf{U}_g = \mathbf{Z}_g^{\text{c}} - \mathbf{Z}_g^{\text{n}}. \tag{59}$$

The global Flow Matching objective is

$$\mathcal{L}_{\text{FM}}^g = \mathbb{E}_t\left[\left\|\mathbf{v}_\theta^g(\mathbf{Z}_t^g, t \mid \mathbf{c}_g) - \mathbf{U}_g\right\|_2^2\right]. \tag{60}$$

Local velocity regression. The clean local latent target is spatially aligned with the pre-repaired initialization before interpolation, following Appendix D.8. Denoting the aligned target by $\mathbf{Z}_\ell^{\text{tar}}$, the local path and target velocity are

$$\mathbf{Z}_t^\ell = (1-t)\widetilde{\mathbf{Z}}_\ell^{\text{n}} + t\mathbf{Z}_\ell^{\text{tar}}, \qquad \mathbf{U}_\ell = \mathbf{Z}_\ell^{\text{tar}} - \widetilde{\mathbf{Z}}_\ell^{\text{n}}. \tag{61}$$

The local velocity field is evaluated with the same conditioning used in Sec. 4.3,

$$\mathbf{v}_\theta^\ell = \mathbf{v}_\theta^\ell\left(\mathbf{Z}_t^\ell, t \mid \mathbf{Z}_t^g, \mathbf{c}_g, \mathbf{C}\right). \tag{62}$$

Because each local latent contains a three-dimensional spatial anchor and one associated latent feature, the two parts are weighted separately,

$$\begin{aligned}\mathcal{L}_{\text{FM}}^\ell = \mathbb{E}_t\Big[&\lambda_{\text{xyz}}\left\|\mathbf{v}_{\theta,\text{xyz}}^\ell(\mathbf{Z}_t^\ell, t \mid \mathbf{Z}_t^g, \mathbf{c}_g, \mathbf{C}) - \mathbf{U}_{\text{xyz}}^\ell\right\|_2^2 \\ &+ \lambda_{\text{feat}}\left\|\mathbf{v}_{\theta,\text{feat}}^\ell(\mathbf{Z}_t^\ell, t \mid \mathbf{Z}_t^g, \mathbf{c}_g, \mathbf{C}) - \mathbf{U}_{\text{feat}}^\ell\right\|_2^2\Big],\end{aligned} \tag{63}$$

with $\lambda_{\text{xyz}} = 1.0$, $\lambda_{\text{feat}} = 0.05$. The complete hierarchical velocity-regression objective is

$$\boxed{\mathcal{L}_{\text{FM}} = \lambda_g\mathcal{L}_{\text{FM}}^g + \lambda_\ell\mathcal{L}_{\text{FM}}^\ell}, \qquad \lambda_g = 0.5, \quad \lambda_\ell = 1.0. \tag{64}$$

E.3 Auxiliary Endpoint Geometry Objectives

Velocity regression constrains the learned transport field at sampled times. We additionally construct training-only extrapolated endpoint estimates from the velocity predicted at time t. These estimates are used only to define auxiliary endpoint losses. They are not the inference trajectory, which is obtained by integrating the learned velocity fields with the multi-step solver described in Sec. 4.3.

The extrapolated global and local endpoints are $\overline{\mathbf{Z}}_g = \mathbf{Z}_t^g + (1-t)\mathbf{v}_\theta^g(\mathbf{Z}_t^g, t \mid \mathbf{c}_g)$,

$$\overline{\mathbf{Z}}_\ell = \mathbf{Z}_t^\ell + (1-t)\mathbf{v}_\theta^\ell(\mathbf{Z}_t^\ell, t \mid \mathbf{Z}_t^g, \mathbf{c}_g, \mathbf{C}). \tag{65}$$

The global endpoint term is $\mathcal{L}_{\text{end}}^g = \left\|\overline{\mathbf{Z}}_g - \mathbf{Z}_g^{\text{c}}\right\|_2^2$.

Directional geometric operators. For two point sets A and B, define the mean directional squared nearest-neighbor discrepancy $D_{A\to B} = \frac{1}{|A|}\sum_{\mathbf{a}\in A}\min_{\mathbf{b}\in B}\|\mathbf{a}-\mathbf{b}\|_2^2$, and the symmetric Chamfer term $\mathcal{C}(A,B) = D_{A\to B} + D_{B\to A}$. For the directional squared-distance set associated with $A \to B$, let $T_{A\to B}^{(q)}$ denote the mean of the largest fraction q of distances. We use $q = 0.20$ for coverage-tail terms and $q = 0.10$ for prediction-outlier terms.

Latent endpoint geometry. Let $\overline{\mathcal{A}}_\ell$ and $\mathcal{A}_\ell^{\mathrm{tar}}$ denote the spatial-anchor channels of $\overline{\mathbf{Z}}_\ell$ and $\mathbf{Z}_\ell^{\mathrm{tar}}$, respectively. These quantities remain in the standardized local-latent coordinate system used by GLFM. We sample 256 anchors from each set and compute

$$\begin{aligned}\mathcal{L}_{\mathrm{lat}}^{\mathrm{CD}} &= \mathcal{C}(\overline{\mathcal{A}}_\ell, \mathcal{A}_\ell^{\mathrm{tar}}),\\ \mathcal{L}_{\mathrm{lat}}^{\mathrm{cov}} &= D_{\mathcal{A}_\ell^{\mathrm{tar}} \to \overline{\mathcal{A}}_\ell},\\ \mathcal{L}_{\mathrm{lat}}^{\text{cov-tail}} &= T^{(0.20)}_{\mathcal{A}_\ell^{\mathrm{tar}} \to \overline{\mathcal{A}}_\ell},\\ \mathcal{L}_{\mathrm{lat}}^{\mathrm{out}} &= T^{(0.10)}_{\overline{\mathcal{A}}_\ell \to \mathcal{A}_\ell^{\mathrm{tar}}}.\end{aligned} \tag{66}$$

The latent endpoint objective is

$$\begin{aligned}\mathcal{L}_{\mathrm{lat}} = a_{\mathrm{den}}(e)\left(0.1\mathcal{L}_{\mathrm{end}}^{g} + 0.5\mathcal{L}_{\mathrm{lat}}^{\mathrm{CD}}\right)\\ + a_{\mathrm{cov}}(e)\left(0.2\mathcal{L}_{\mathrm{lat}}^{\mathrm{cov}} + 0.1\mathcal{L}_{\mathrm{lat}}^{\text{cov-tail}}\right)\\ + 0.05 a_{\mathrm{out}}(e)\mathcal{L}_{\mathrm{lat}}^{\mathrm{out}}.\end{aligned} \tag{67}$$

Decoded point-cloud geometry. The extrapolated endpoint latents are first transformed back to the decoder latent scale and then decoded as $\overline{\mathbf{X}}_{\mathrm{p}} = D(\overline{\mathbf{Z}}_g, \overline{\mathbf{Z}}_\ell)$. Using 256 predicted and clean points during training, we compute

$$\begin{aligned}\mathcal{L}_{\mathrm{dec}}^{\mathrm{CD}} &= \mathcal{C}(\overline{\mathbf{X}}^{\mathrm{p}}, \mathbf{X}^{\mathrm{c}}),\\ \mathcal{L}_{\mathrm{dec}}^{\mathrm{cov}} &= D_{\mathbf{X}^{\mathrm{c}} \to \overline{\mathbf{X}}^{\mathrm{p}}},\\ \mathcal{L}_{\mathrm{dec}}^{\text{cov-tail}} &= T^{(0.20)}_{\mathbf{X}^{\mathrm{c}} \to \overline{\mathbf{X}}^{\mathrm{p}}},\\ \mathcal{L}_{\mathrm{dec}}^{\mathrm{out}} &= T^{(0.10)}_{\overline{\mathbf{X}}^{\mathrm{p}} \to \mathbf{X}^{\mathrm{c}}}.\end{aligned} \tag{68}$$

The decoded endpoint objective is

$$\mathcal{L}_{\mathrm{dec}} = a_{\mathrm{dec}}(e)\left[\mathcal{L}_{\mathrm{dec}}^{\mathrm{CD}} + 0.5\mathcal{L}_{\mathrm{dec}}^{\mathrm{cov}} + 0.2\mathcal{L}_{\mathrm{dec}}^{\text{cov-tail}} + 0.1\mathcal{L}_{\mathrm{dec}}^{\mathrm{out}}\right]. \tag{69}$$

During GLFM training, a micro-batch denotes the 32-sample forward batch processed before gradient accumulation. The decoded endpoint objective is evaluated once every eight such micro-batches. On evaluated micro-batches, its contribution is multiplied by eight so that its expected weight matches evaluation at every micro-batch while reducing decoder-side training cost.

E.4 Complete Restoration Objective and Loss Scheduling

The GLFM stage is optimized with the objective introduced in Sec. 4.4:

$$\boxed{\mathcal{L}_{\mathrm{rest}} = \lambda_{\mathrm{FM}}\mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{lat}}\mathcal{L}_{\mathrm{lat}} + \lambda_{\mathrm{dec}}\mathcal{L}_{\mathrm{dec}}}. \tag{70}$$

Here, $\lambda_{\mathrm{FM}} = \lambda_{\mathrm{lat}} = \lambda_{\mathrm{dec}} = 1$. The coefficients in equations 64, 67, and 69 determine the relative contributions of the individual terms.

The auxiliary endpoint terms are introduced progressively. For epoch e, start epoch s, and warm-up length w, define

$$A(e; s, w) = \begin{cases} 0, & e < s,\\ \min\left(1, \dfrac{e - s + 1}{w}\right), & e \geq s.\end{cases} \tag{71}$$

We use $a_{\mathrm{den}}(e) = A(e; 1, 5)$, $a_{\mathrm{out}}(e) = A(e; 6, 5)$, $a_{\mathrm{cov}}(e) = A(e; 11, 10)$, $a_{\mathrm{dec}}(e) = A(e; 21, 10)$. Velocity-field supervision $\mathcal{L}_{\mathrm{FM}}$ is active throughout GLFM training. Base latent endpoint supervision is ramped from the first epoch, while outlier, coverage, and decoded-geometry terms are introduced progressively at later stages.

E.5 Optimization Protocol

GALR optimization. The hierarchical autoencoder is first trained on clean 2,048-point target clouds. Its encoder E and decoder D are then frozen. GALR is optimized with AdamW for 50

Table 10: Loss coefficients used for GALR and Global-Local Flow Matching.

Term	Coefficient
GALR calibration λ_{cal}	0.2
Focal positive factor α_{f}	0.75
Focal exponent γ_{f}	2.0
Structure-weight extra weight λ_{str}	2.0
Global FM λ_g	0.5
Local FM λ_ℓ	1.0
Local spatial λ_{xyz}	1.0
Local feature λ_{feat}	0.05
Global endpoint	0.1
Latent Chamfer	0.5
Latent coverage	0.2
Latent coverage tail	0.1
Latent outlier tail	0.05
Decoded Chamfer	1.0
Decoded coverage	0.5
Decoded coverage tail	0.2
Decoded outlier tail	0.1
Coverage-tail fraction	0.20
Outlier-tail fraction	0.10

epochs using a batch size of 64, a learning rate of 1×10^{-5}, and a weight decay of 1×10^{-4}. Gradients are clipped at norm 1.0, and validation is performed after every epoch.

For checkpoint selection, the unchanged noisy global latent and the pre-repaired local latent are first returned to the decoder latent scale. For notational simplicity, we denote the resulting decoded pre-repair output as $\mathbf{X}_{\mathrm{pre}} = D\left(\mathbf{Z}_g^{\mathrm{n}}, \widetilde{\mathbf{Z}}_\ell^{\mathrm{n}}\right)$. Using the directional operators defined above, we compute $C_{\mathrm{dec}} = \mathcal{C}(\mathbf{X}_{\mathrm{pre}}, \mathbf{X}_{\mathrm{c}})$,

$$T_{\mathrm{cov}} = T^{(0.20)}_{\mathbf{X}^{\mathrm{c}} \to \mathbf{X}_{\mathrm{pre}}}, \qquad T_{\mathrm{out}} = T^{(0.10)}_{\mathbf{X}_{\mathrm{pre}} \to \mathbf{X}^{\mathrm{c}}}. \tag{72}$$

The GALR validation score is $S_{\mathrm{GALR}} = C_{\mathrm{dec}} + 0.5T_{\mathrm{cov}} + 0.5T_{\mathrm{out}}$, and the checkpoint with the lowest validation score is retained.

Flow Matching optimization. After GALR is frozen, the global and local velocity networks are optimized with AdamW for 80 epochs. We use a micro-batch size of 32 and accumulate gradients over four micro-batches, giving an effective batch size of 128. The initial learning rate is 5×10^{-5}, the minimum learning rate is 5×10^{-6}, and the weight decay is 1×10^{-4}. The first five epochs use linear learning-rate warm-up, followed by cosine decay. Gradients are clipped at norm 1.0. Training uses mixed precision and TF32 acceleration on a single NVIDIA RTX PRO 6000 GPU.

Periodic validation is performed every five epochs on a fixed category-stratified subset containing 512 validation objects. The selection criterion is the macro-average decoded Chamfer error across the Low, Mid, and High corruption groups. The three best periodic checkpoints are subsequently re-evaluated on the complete validation set, and the checkpoint with the lowest full-validation criterion is used for test evaluation.

F Baseline Adaptation and Re-Training Protocols

This section documents how each baseline is adapted and retrained on SPORE. The protocol preserves each method's official architecture and core training recipe where compatible, while standardizing dataset splits, point cardinality, evaluation, and validation-based model selection.

F.1 Common Fair-Comparison Protocol

All learning-based baselines are retrained from scratch on the same mixed-noise SPORE training split used by PRISM. The split contains 41,880 ShapeNet objects and 125,495 available noisy–clean training pairs across the Low, Mid, and High corruption regimes. A single model is trained for each baseline and is evaluated separately on the three test regimes. The validation and test object partitions are fixed across all methods.

Table 11: Optimization configuration for GALR and Global-Local Flow Matching.

Setting	Configuration
GALR epochs	50
GALR batch size	64
GALR optimizer	AdamW
GALR learning rate	1×10^{-5}
GALR weight decay	1×10^{-4}
Flow Matching epochs	80
FM micro-batch size	32
Gradient accumulation	4 micro-batches
Effective FM batch size	128
FM optimizer	AdamW
Initial FM learning rate	5×10^{-5}
Minimum FM learning rate	5×10^{-6}
FM learning-rate schedule	5-epoch warm-up + cosine decay
Gradient clipping	1.0
Decoded-loss interval	8 micro-batches
Validation interval	5 epochs
Validation subset	512 objects
Hardware	1× NVIDIA RTX PRO 6000

Adaptation policy. We preserve each baseline's official architecture and core optimization recipe whenever they are compatible with SPORE. Dataset-dependent components, including the data interface, point cardinality, coordinate normalization, patch sampling, and output aggregation, are adapted consistently to the benchmark. No test-set metric is used for architecture selection, hyperparameter selection, checkpoint selection, normalization, inference-parameter selection, or post-processing design. If memory constraints require a smaller micro-batch, gradient accumulation is used to preserve the intended effective batch size.

Input-defined normalization. Normalization statistics are computed from the noisy observation only. The resulting translation and scale are applied to both members of a noisy–clean pair during training. At inference, the transformation therefore depends only on the noisy input. Method-specific local patch normalization is retained when required by the original implementation, but its statistics are computed from the noisy patch.

Validation-based model selection. To use a common final model-selection rule, we evaluate complete object-level predictions on the Low, Mid, and High validation groups. The final restoration checkpoint is selected by the macro-average validation CD-L2 across the three regimes. P2M, F@0.01, HD95, and all test-set results are excluded from checkpoint selection. For multi-stage methods, intermediate components are selected using their native validation objective, whereas the final restoration model is selected by macro-average validation CD-L2.

Training budget and epoch definition. Each method is assigned the maximum training budget in Table 12. The budgets follow the official implementation or published training recipe, while retaining a larger budget when our SPORE adaptation is deliberately more generous. Validation is performed at fixed intervals throughout training, and the checkpoint with the best validation criterion is retained for final evaluation; the last training epoch is not used by default. For IterativePFN, PD-LTS, ASDN, and PGD, one SPORE epoch is defined as one on-the-fly randomly sampled patch from each of the 125,495 available noisy–clean training pairs. PointCleanNet follows a separate patch-sampling schedule described below. The test set is evaluated only after training and validation-based model selection are complete.

Patch aggregation and output cardinality. Each benchmark observation and clean target contains 2,048 points. We retain the native patch-fusion or patch-stitching procedure when a baseline defines one. Otherwise, predicted patches are mapped back to the benchmark coordinate frame before aggregation. Multiple estimates of the same source point are averaged when point-wise correspondence is available, whereas unordered patch outputs are merged before cardinality restoration. Outputs with more than 2,048 points are reduced by farthest-point sampling, while outputs with fewer than 2,048 points are restored by fixed local interpolation with $K_{\mathrm{interp}} = 8$. These rules are fixed before test evaluation and are not changed across corruption regimes.

Inference hyperparameters. When an official method exposes a small discrete set of inference-iteration choices, the iteration count is selected on the validation split only. For each corruption regime $r \in \{\mathrm{Low}, \mathrm{Mid}, \mathrm{High}\}$, we select $k_r^\star = \arg\min_{k \in \mathcal{K}} \mathrm{CD}_{\mathrm{val},r}(k)$, where $\mathcal{K}$ is restricted to iteration counts documented by the official implementation. The selected iteration count is then frozen for the corresponding test regime and is used unchanged to compute all four reported metrics. No metric-specific inference setting is selected.

All retraining and evaluation experiments use a single NVIDIA RTX PRO 6000 GPU. Final predictions are expressed in the benchmark coordinate frame and contain 2,048 points before evaluation. We use the metric definitions in Appendix C: CD-L2, bidirectional P2M, F@0.01, and HD95.

F.2 Learning-Based Baseline Protocols

PointCleanNet. PointCleanNet retains the ResPCPNet restoration backbone and its official patch geometry. Each local patch contains 500 points within a radius equal to 0.05 times the object bounding-box diagonal. We use SGD with learning rate 1×10^{-8}, momentum 0.9, and batch size 64, following the official training implementation. Because SPORE contains substantially more training objects than the original dataset, one benchmark epoch is defined as 102,400 randomly sampled patches from the complete mixed-noise training pool. Training is allowed to run for up to 2,000 such epochs, with validation every 10 epochs. Inference follows the official full-point correction procedure together with neighborhood displacement aggregation, using 100 neighbors for the aggregation step, and preserves the 2,048-point output cardinality.

Score-Based Point Cloud Denoising. The score-based baseline uses 1,000-point local patches and paired geometric augmentation. We retain Adam with learning rate 1×10^{-4}, zero weight decay, batch size 32, and a maximum budget of 1,000,000 optimization steps. Validation is performed every 2,000 steps. The Langevin denoiser uses 30 inner steps, initial step size 0.2, decay factor 0.95, and a four-neighbor score ensemble ($K_{\mathrm{denoise}} = 4$). The official object-level protocol uses one or two outer denoising iterations depending on noise severity. We therefore set $\mathcal{K} = \{1, 2\}$ and select the outer iteration count for each SPORE validation regime using validation CD-L2 only. Native patch-based denoising and aggregation are retained.

PD-Flow. PD-Flow uses 1,024-point patches and retains its normalizing-flow restoration objective. Optimization uses Adam with learning rate 2×10^{-3} and batch size 8. The loss weights are 0.1 for reconstruction, 10^{-6} for log-likelihood, and 10.0 for consistency. We train for up to 700,000 optimizer updates and monitor denoised validation CD throughout training. Gradient clipping is set to 10^{-3}. Patch predictions are merged using the method's native procedure before cardinality restoration.

P2P-Bridge. P2P-Bridge uses the official object-level PVD-large configuration with 2,048-point inputs. We retain AdamW with learning rate 3×10^{-4}, weight decay 1×10^{-5}, $(\beta_1, \beta_2) = (0.9, 0.999)$, batch size 32, gradient clipping at 1.0, mixed-precision training, and exponential moving averaging with decay 0.999. The maximum training budget is 450,000 optimization steps, following the official PU-Net configuration. The diffusion objective predicts noise with mean-squared error, and test-time restoration uses 10 DDPM sampling steps.

IterativePFN. IterativePFN uses 1,000-point local patches and its native iterative filtering architecture. We retain Adam with learning rate 1×10^{-4} and batch size 4. ReduceLROnPlateau uses patience 10, reduction factor 0.5, and minimum learning rate 1×10^{-7}. Training is allowed to run for up to 100 full-data epochs, matching the official implementation. Under the SPORE epoch definition above, each epoch contains 125,495 on-the-fly training patches, one sampled from each available noisy–clean pair. Validation is performed after every epoch, and inference uses the trained internal iterative filtering process without an additional external refinement schedule.

PD-LTS. We use the official PD-LTS light configuration for object-level denoising. Training uses 1,024-point patches, Adam with learning rate 2×10^{-3}, and batch size 16. ReduceLROnPlateau uses patience 5, reduction factor 0.5, and minimum learning rate 1×10^{-6}. The official implementation trains for 150 epochs with on-the-fly patch sampling. Because SPORE contains substantially more

training observations, we use the common SPORE epoch definition above, yielding 125,495 on-the-fly training patches per epoch, and allow a maximum budget of 200 epochs. The final checkpoint is selected by macro-average validation CD-L2 across Low, Mid, and High. Inference uses one denoising iteration, consistent with the official object-level protocol, and retains the native patch-based denoising procedure.

ASDN. ASDN follows its official two-stage optimization procedure. The local scale classifier is trained first with Adam, learning rate 1×10^{-4}, batch size 12, and a maximum budget of 800 epochs. The denoising network is then trained with the selected classifier fixed, using Adam, learning rate 5×10^{-4}, batch size 24, and a maximum budget of 800 epochs. Both stages use 1,000-point patches and their official ReduceLROnPlateau schedules. Under the common SPORE epoch definition, each stage receives 125,495 on-the-fly training patches per epoch. For inference, the official implementation uses two or three restoration iterations depending on noise severity. We therefore set $\mathcal{K} = \{2, 3\}$ and select the iteration count independently for each SPORE validation regime using validation CD-L2 only. Native patch-based restoration is retained.

PGD. PGD uses 1,000-point local patches and retains its structural-prior-guided restoration objective. We use Adam with learning rate 5×10^{-4} and batch size 20. ReduceLROnPlateau uses patience 2, reduction factor 0.5, and minimum learning rate 1×10^{-9}. Training is allowed to run for up to 800 epochs, matching the official implementation. Under the common SPORE epoch definition, each epoch contains 125,495 on-the-fly training patches. The official inference protocol uses two or three restoration iterations according to noise severity. We therefore set $\mathcal{K} = \{2, 3\}$ and select the iteration count independently for each SPORE validation regime using validation CD-L2 only. Native patch-based restoration is retained.

F.3 Statistical Outlier Removal

Statistical Outlier Removal (SOR) is the non-learning baseline. For each input point, we compute the mean Euclidean distance d_i to its K nearest neighbors. Let μ_d and σ_d denote the mean and standard deviation of $\{d_i\}$. A point is retained when $d_i \leq \mu_d + \beta\sigma_d$. The SOR hyperparameters are selected exclusively on the validation split using macro-average CD-L2 across the Low, Mid, and High regimes. We evaluate $K \in \{16, 32, 64\}, \beta \in \{1.0, 1.5, 2.0, 2.5, 3.0\}$ and obtain the validation-optimal configuration $K = 32, \beta = 2.0$. P2M, F@0.01, HD95, and all test-set results are excluded from this selection. The minimum retained cardinality and interpolation neighborhood are fixed independently of test performance: we require at least 1,536 retained points, corresponding to 75% of the 2,048-point input, and use $K_{\text{interp}} = 8$ for cardinality restoration. If thresholding retains fewer than 1,536 points, rejected points are ranked by increasing d_i and restored until 1,536 points remain. The filtered cloud is then returned to 2,048 points with the fixed local interpolation operator. The complete SOR procedure is frozen before test evaluation.

F.4 Re-Training Configuration Summary

Table 12: Maximum re-training budgets used for the final baseline comparison. Final checkpoints and, where applicable, inference iteration counts are selected exclusively on the validation split. No test-set result is used for model or inference-parameter selection.

Method	Optimizer	Batch	Maximum budget	Validation / inference
PointCleanNet	SGD	64	2,000 ep.	val/10 ep.; native correction
Score-Based	Adam	32	1M steps	val/2k; $\{1, 2\}$ outer iters
PD-Flow	Adam	8	700k steps	validation CD; native fusion
P2P-Bridge	AdamW	32	450k steps	EMA 0.999; 10 DDPM steps
IterativePFN	Adam	4	100 ep.	val every ep.; native iteration
PD-LTS	Adam	16	200 ep.	validation CD; 1 iter.
ASDN classifier	Adam	12	800 ep.	native validation objective
ASDN denoiser	Adam	24	800 ep.	validation CD; $\{2, 3\}$ iters
PGD	Adam	20	800 ep.	validation CD; $\{2, 3\}$ iters
SOR	–	–	validation grid	$K = 32, \beta = 2.0$

G Full Ablation Results and Exact Variant Definitions

The main paper reports ablations under High single-photon degradation. Here we provide the corresponding Low- and Mid-degradation results and define each intervention precisely. We use the same four metrics as in the main paper: CD-L2 and P2M are reported in units of 10^{-3}, whereas F@0.01 and HD95 follow Appendix C. The High-degradation results remain in Table 2 and are not duplicated here.

The ablations are organized hierarchically. Framework-level variants assess GALR and GLFM and separate repaired initialization from persistent GALR-derived conditioning. GLFM design variants then test two local mechanisms that use the retained repair information: reliability-aware neighborhood interaction and point-wise FiLM. Each comparison is interpreted only within its stated intervention.

G.1 Common Ablation Protocol

All ablation variants use the same SPORE training, validation, and test partitions as the full model and are evaluated with the same metric implementation. The hierarchical autoencoder is fixed across all variants. When GALR is retained, the same frozen GALR checkpoint is used. Unless an intervention explicitly removes or changes a component, GLFM variants use the same architecture depth, optimizer, learning-rate schedule, training budget, validation-based checkpoint selection, four-step Euler solver, and evaluation pipeline as the full configuration. Differences between a variant and the full model are therefore interpreted only within the scope of the stated intervention.

G.2 Exact Variant Definitions

Full model. The full configuration applies GALR before GLFM. GLFM is initialized from the repaired local latent $\widetilde{\mathbf{Z}}_\ell^{\mathrm{n}}$ and retains the GALR-derived object-level context $\mathbf{c}_g$ and token-wise repair-state set $\mathbf{C}$. The local branch uses the reliability-aware $K = 16$ neighborhood defined in Sec. 4.3 and point-wise FiLM throughout the four-layer KNN Transformer.

GALR only (w/o GLFM). This variant evaluates geometry-aware latent repair without subsequent latent transport. GALR first produces the repaired local latent $\widetilde{\mathbf{Z}}_\ell^{\mathrm{n}}$, after which the unchanged noisy global latent and repaired local latent are decoded directly, $\mathbf{X}^{\mathrm{p}} = D\left(\mathbf{Z}_g^{\mathrm{n}}, \widetilde{\mathbf{Z}}_\ell^{\mathrm{n}}\right)$. No Global-Local Flow Matching is performed. This variant therefore measures what GALR alone recovers before learned latent transport.

w/o GALR. GALR is removed while the GLFM backbone is retained. GLFM therefore starts from the original noisy local latent $\mathbf{Z}_\ell^{\mathrm{n}}$ rather than $\widetilde{\mathbf{Z}}_\ell^{\mathrm{n}}$ and receives no GALR-derived conditioning. Apart from the removed GALR-derived inputs, the remaining GLFM architecture and training protocol are retained. Because this intervention removes both GALR-based pre-repair and GALR-derived conditioning, comparison with the full model measures the framework-level contribution of GALR as a whole rather than the effect of repaired initialization alone.

w/o Repair-State Conditioning. This variant retains GALR and initializes GLFM from the same repaired local latent $\widetilde{\mathbf{Z}}_\ell^{\mathrm{n}}$ as the full model, but removes both forms of persistent GALR-derived conditioning during latent transport. Specifically, the object-level context $\mathbf{c}_g$ and token-wise repair-state set $\mathbf{C}$ are both removed from the GLFM conditioning inputs. The repaired starting representation and the remaining GLFM backbone are unchanged. This comparison separates the benefit of repaired initialization from the additional contribution of the complete persistent repair-conditioning pathway. It should not be interpreted as an isolated ablation of $\mathbf{c}_g$ or $\mathbf{C}$ individually.

w/o Reliability-aware Neighborhood. GALR, the repaired initialization, repair-state conditioning, the four-layer local Transformer, and point-wise FiLM are retained. The neighborhood size remains $K = 16$, but reliability is removed from both neighborhood construction and attention. Specifically, the reliability-adjusted distance is replaced by the ordinary Euclidean distance, $\widetilde{d}_{ij}(t) \to \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|_2$, and the reliability-dependent attention bias is disabled, $b_{ij}^{\mathrm{rel}} = 0$. This

comparison isolates the contribution of reliability-aware local neighborhood interaction while retaining local message passing itself.

w/o Point-wise FiLM. The repaired initialization, object-level context, token-wise repair information, and reliability-aware $K = 16$ neighborhood are retained, but token-wise FiLM modulation is disabled in the local Transformer. Equivalently, the point-dependent modulation terms are removed while the remaining global and local conditioning pathways are unchanged. This variant isolates the contribution of spatially resolved FiLM modulation during local latent transport.

Interpretation of the hierarchy. The framework-level variants answer three distinct questions: what GALR contributes at the framework level, whether GALR alone is sufficient, and whether repaired initialization alone accounts for the full gain. The GLFM design variants then examine two local mechanisms through which retained repair information affects local transport. These comparisons isolate reliability-aware neighborhood interaction and point-wise FiLM while preserving the remainder of the full configuration.

G.3 Low-Degradation Ablation

Under Low degradation, configurations that retain both GALR initialization and GLFM remain close in CD-L2 and F@0.01 at the reported precision. The full model achieves the lowest P2M and HD95 among the ablation configurations, whereas removing repair-state conditioning yields a marginally lower CD-L2 and a marginally higher F@0.01. The clearest advantage of the complete conditioning design in this regime therefore appears in HD95.

Removing GALR produces a substantially larger increase in CD-L2 and P2M. Because this intervention removes both pre-repair and GALR-derived conditioning, the result supports a framework-level contribution of GALR under mild corruption but does not isolate repaired initialization alone. That distinction is provided by the repair-conditioning ablation, which retains the same repaired starting latent while removing persistent conditioning. Removing reliability-aware neighborhood interaction or point-wise FiLM has little effect on CD-L2 and F@0.01 in this regime, although both variants increase HD95 from 0.0279 to approximately 0.039. Thus, the internal local-conditioning mechanisms are less distinguishable when the input geometry is already relatively reliable.

G.4 Mid-Degradation Ablation

The Mid regime separates the framework-level configurations more clearly. Removing GALR increases CD-L2 from 1.174 to 4.919 and P2M from 1.242 to 4.083, producing the largest deterioration in these two distance-based metrics among the framework-level variants. Because this intervention removes both GALR-based pre-repair and GALR-derived conditioning, the comparison supports the framework-level contribution of GALR rather than isolating the effect of repaired initialization alone. GALR alone exhibits a different error profile: its P2M of 1.302 and HD95 of 0.0274 remain relatively close to the full model, whereas CD-L2 remains higher and F@0.01 is substantially lower. This pattern suggests that GALR already suppresses large residual errors and recovers substantial surface consistency, while GLFM remains important for overall point-set fidelity and strict local correspondence.

The repair-conditioning ablation provides the controlled separation between repaired initialization and persistent conditioning. GLFM starts from the same repaired latent as the full model, but both $\mathbf{c}_g$ and $\mathbf{C}$ are removed during transport. Under this intervention, CD-L2 increases from 1.174 to 1.434, F@0.01 decreases from 0.2143 to 0.1346, and HD95 increases from 0.0237 to 0.0428. Repaired initialization therefore explains only part of the restoration gain. Persistent GALR-derived conditioning provides additional benefit, particularly for strict correspondence and upper-tail geometric error.

The GLFM-design ablations show a similar severity-dependent separation. Removing reliability-aware neighborhood interaction increases CD-L2 from 1.174 to 1.525 and reduces F@0.01 to 0.1267, while HD95 increases to 0.0431. Removing point-wise FiLM produces a smaller change in the average-distance metrics but reduces F@0.01 to 0.1326 and increases HD95 to 0.0429. Together with the High-degradation results in Table 2, these results indicate that the internal GLFM mech-

Table 13: Hierarchical ablation study under Low single-photon degradation. CD-L2 and P2M are reported in units of 10^{-3}. The upper block compares framework-level configurations, while the lower block ablates internal GLFM designs from the full framework. Check marks denote components or architectural pathways retained in each configuration; their conditioning inputs follow the exact variant definitions in Appendix G.2.

Framework			**GLFM design**		**Metrics**			
GALR	GLFM	Repair State	Reliability-aware Neighborhood	FiLM	CD-L2↓	P2M↓	F@0.01↑	HD95↓
Framework-level ablations								
✓					1.183	1.256	0.1970	0.0393
	✓		✓	✓	3.511	3.190	0.1999	0.0390
✓	✓		✓	✓	**1.162**	1.250	**0.2000**	0.0392
✓	✓	✓	✓	✓	1.164	**1.240**	0.1997	**0.0279**
GLFM design ablations								
✓	✓	✓		✓	1.164	1.246	0.1989	0.0391
✓	✓	✓	✓		1.164	1.250	0.1996	0.0392
✓	✓	✓	✓	✓	1.164	**1.240**	0.1997	**0.0279**

Table 14: Hierarchical ablation study under Mid single-photon degradation. Metric conventions, component columns, and variant definitions are identical to Table 13.

Framework			**GLFM design**		**Metrics**			
GALR	GLFM	Repair State	Reliability-aware Neighborhood	FiLM	CD-L2↓	P2M↓	F@0.01↑	HD95↓
Framework-level ablations								
✓					1.612	1.302	0.1251	0.0274
	✓		✓	✓	4.919	4.083	0.1288	0.0493
✓	✓		✓	✓	1.434	1.253	0.1346	0.0428
✓	✓	✓	✓	✓	**1.174**	**1.242**	**0.2143**	**0.0237**
GLFM design ablations								
✓	✓	✓		✓	1.525	1.291	0.1267	0.0431
✓	✓	✓	✓		1.436	1.251	0.1326	0.0429
✓	✓	✓	✓	✓	**1.174**	**1.242**	**0.2143**	**0.0237**

anisms become substantially more consequential under Mid and High degradation, particularly for strict local correspondence and control of large residual errors.

Cross-regime summary. Across Low, Mid, and High degradation, the framework-level comparisons support complementary roles for GALR and GLFM, while the repair-conditioning ablation shows that repaired initialization alone does not account for the full model. The GLFM-design comparisons further support distinct contributions from reliability-aware neighborhood interaction and point-wise FiLM. Their effects are most pronounced under Mid and High degradation, particularly in F@0.01 and HD95, rather than increasing monotonically for every metric. Together, these observations support the intended PRISM information pathway: GALR first repairs unreliable local representations and produces repair information that is retained throughout GLFM, while reliability-aware neighborhood interaction and point-wise FiLM provide two local mechanisms for exploiting this information during restoration.

H Additional Qualitative and Mechanistic Analysis

This section complements the main quantitative comparison and Appendix G with qualitative results across all three corruption regimes, a sample-level GALR visualization, and representative failure cases. The visual comparisons use selected baselines for readability, while the main paper reports the complete numerical comparison. The mechanism figure visualizes estimated reliability, derived repair weights, realized anchor corrections, and pre-repaired geometry. The failure cases highlight localized errors that remain challenging for the current pipeline.

H.1 Qualitative Comparisons across Corruption Regimes

CD-L2 and P2M follow the main-paper convention and are reported in units of 10^{-3}.

Low degradation. Under Low degradation, the quantitative ranking is mixed because several methods already achieve small average geometric errors. PGD, PD-LTS, and PD-Flow obtain lower CD-L2 than PRISM, while PGD and PD-LTS also obtain lower P2M. PRISM achieves the lowest HD95 of 0.0279 and ties SOR for the highest F@0.01 of 0.1997. Figure 6 is consistent with this close competition. Most methods recover the dominant object geometry, while the visible differences are concentrated in isolated residual points, local surface regularity, and preservation of thin structures. PCN and SOR retain more peripheral responses in several examples, whereas PGD, PD-LTS, and PRISM produce more compact object reconstructions. The visual similarity among the stronger methods is consistent with the relatively small separation in the Low-regime distance metrics.

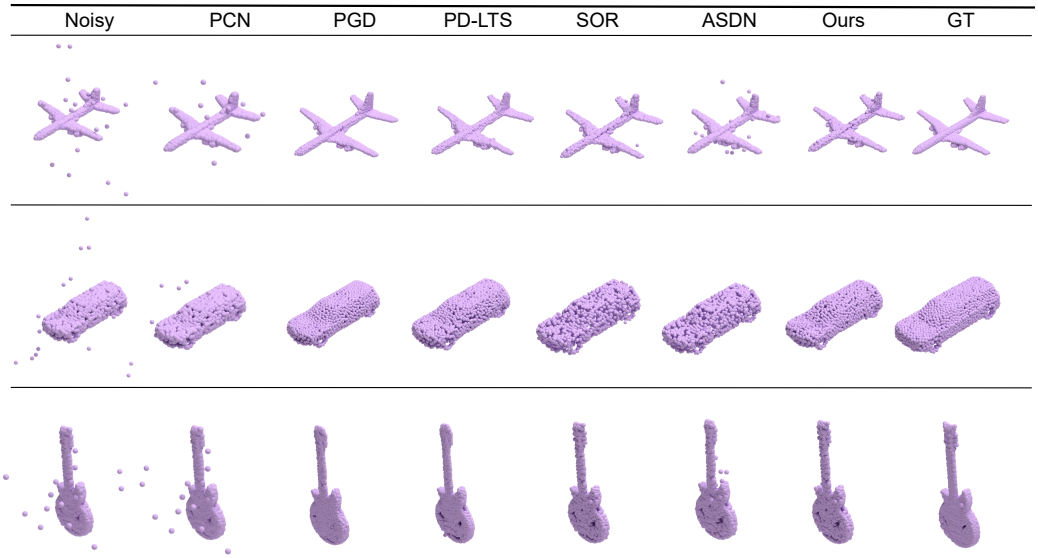


Figure 6: Additional qualitative comparison under Low single-photon degradation. Columns show the noisy observation, selected baselines (PCN, PGD, PD-LTS, SOR, and ASDN), PRISM, and the clean ground truth (GT). Under mild corruption, several methods recover recognizable geometry, consistent with the close quantitative competition in this regime. Differences are most visible in residual isolated points, local surface regularity, and thin-structure preservation.

Mid degradation. The separation becomes clearer under Mid degradation. PRISM achieves the best CD-L2 (1.174), F@0.01 (0.2143), and HD95 (0.0237), while its P2M of 1.242 remains close to the best value of 1.200. Figure 7 shows the corresponding qualitative trend. PCN, SOR, and ASDN retain more spatially isolated responses in several examples. PGD and PD-LTS recover the dominant object structure more effectively, but local irregularities remain visible around thin or weakly supported regions. PRISM produces coherent reconstructions with fewer peripheral outliers while preserving the dominant object geometry, consistent with its stronger $F@0.01$ and HD95 in this regime.

High degradation. High degradation exposes the largest restoration challenge because severe outliers and weak local geometric support occur simultaneously. Quantitatively, PRISM achieves the best result on all four reported metrics, with CD-L2 of 5.248, P2M of 2.154, F@0.01 of 0.3419, and HD95 of 0.0297. Relative to the strongest competing result for each metric, this corresponds to reductions of 2.4% in CD-L2, 18.2% in P2M, and 27.6% in HD95, together with a 22.9% improvement in F@0.01. The additional examples in Fig. 8 are consistent with these quantitative differences. PCN and SOR retain substantial diffuse corruption, while ASDN suppresses part of the isolated responses but leaves incomplete local geometry in several cases. PGD and PD-LTS re-

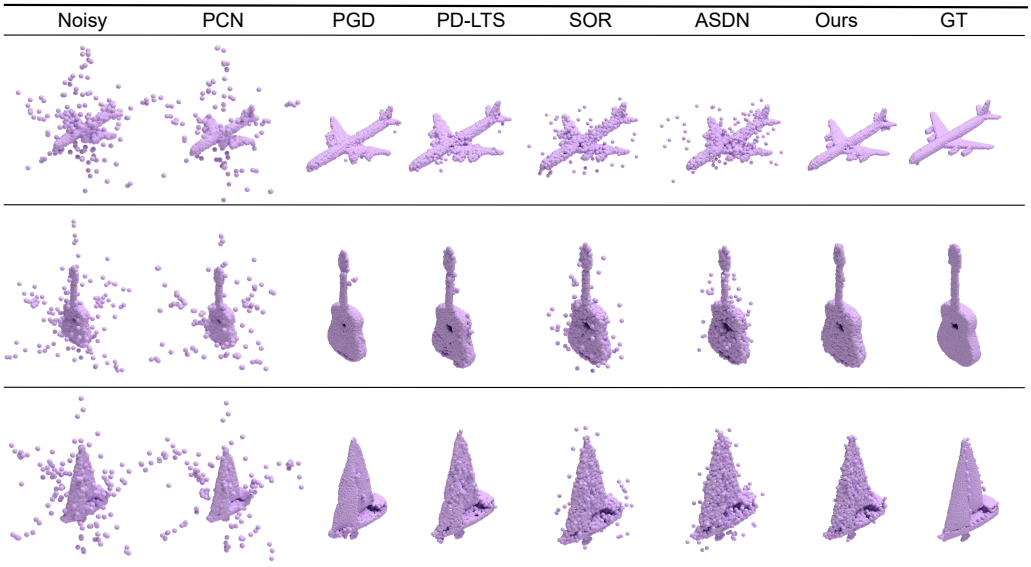


Figure 7: Additional qualitative comparison under Mid single-photon degradation. As corruption increases, differences in residual outliers and local structural continuity become more visible. The comparison is restricted to selected representative baselines for visual clarity; the complete quantitative comparison is reported in the main paper.

cover the dominant object structure more clearly, although residual local clusters or fragmented thin regions remain visible. PRISM more consistently recovers continuous object structure with fewer severe residual points, particularly in the elongated and weakly supported regions represented by these examples.

Across the selected examples, the qualitative differences are consistent with the aggregate quantitative trend: separation among strong methods is modest under Low degradation and becomes more apparent in the Mid and High regimes, particularly in residual outliers and thin-structure continuity.

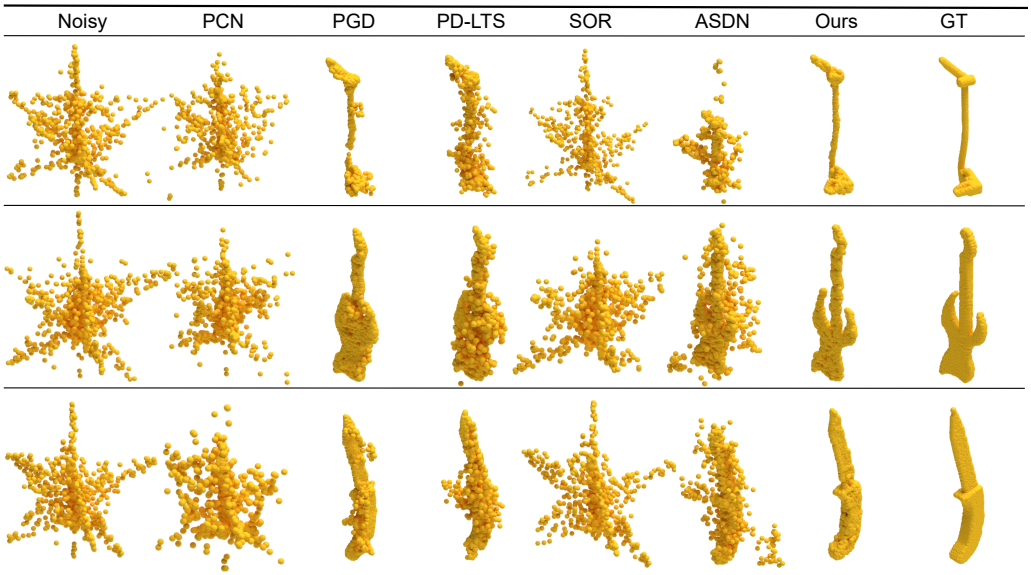


Figure 8: Additional qualitative comparison under High single-photon degradation. These examples complement the High-degradation visualization in the main paper. Severe corruption produces diffuse outliers and weak local support; PRISM suppresses large residual responses while recovering coherent object structure across the shown examples.

H.2 Mechanistic Visualization of Geometry-Aware Latent Repair

Figure 9 visualizes the internal quantities produced by GALR for a severely corrupted example. The noisy observation contains a compact central structure together with strongly displaced and spatially isolated points. The estimated reliability r_i is higher in the coherent central region and generally lower for peripheral or displaced observations. The repair weight w_i is derived monotonically from the calibrated corruption probability and therefore assigns larger weights to lower-reliability regions by construction. The raw repair magnitude $\|\boldsymbol{\delta}_i\|_2$ shows where the realized anchor corrections are largest; the bounded repair-state variable $m_i = \tanh(\|\boldsymbol{\delta}_i\|_2)$ is derived from this raw magnitude. The inset illustrates a representative repair displacement. After applying these corrections, the pre-repaired anchor configuration forms a more coherent intermediate geometry with substantially fewer spatially dispersed responses.

The visualization provides sample-level evidence only; it does not establish a dataset-level causal relation between reliability and displacement. Component-level quantitative evidence is provided by the controlled ablations in Appendix G. Removing GALR or the downstream use of GALR-derived repair information degrades multiple metrics, with clearer separation under Mid and High corruption. These observations are consistent with the proposed pathway in which GALR provides a repaired initialization and explicit repair information before GLFM performs global–local latent transport.

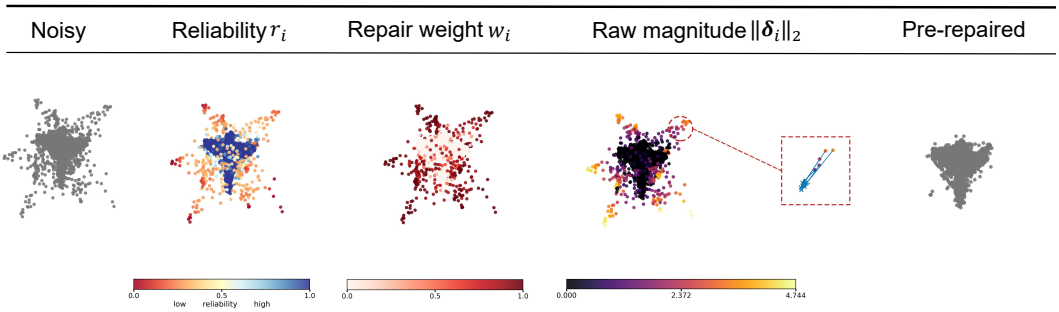


Figure 9: Mechanistic visualization of Geometry-Aware Latent Repair under severe single-photon corruption. From left to right: noisy observation, estimated reliability r_i, repair weight w_i, raw repair magnitude $\|\boldsymbol{\delta}_i\|_2$, and the resulting pre-repaired anchor configuration. Colors follow the panel-specific color bars. Lower reliability maps to larger repair weights by construction, while the raw magnitude visualizes the realized anchor-displacement size. The pre-repaired configuration shows the intermediate geometry passed to subsequent latent transport.

H.3 Failure Cases and Limitations

Figure 10 presents representative failure cases under High single-photon degradation, in which PRISM recovers the dominant object geometry but leaves localized corruption or incompletely restores fine-scale structure. Two representative failure patterns are visible in these examples. First, a small number of residual points remain attached to or immediately adjacent to the reconstructed object. These residuals lie close to plausible surface regions or form compact local clusters, a pattern consistent with ambiguity between near-surface corruption and valid local support. Second, small or weakly supported structures can be smoothed, shortened, or omitted. In the displayed examples, PRISM reconstructs the main elongated body but does not fully recover every fine protrusion or boundary detail visible in the ground truth.

These examples highlight two remaining limitations of the current restoration pipeline: near-surface residual corruption and incomplete recovery of weakly supported fine structures. Localized errors can remain visually salient even when the aggregate restoration metrics are favorable, indicating that CD-L2, P2M, F@0.01, and HD95 summarize complementary aspects of restoration quality without resolving every local geometric detail. Improving the separation of near-surface corruption from valid geometry, together with stronger preservation of weakly supported fine structures, remains an important direction for future work.

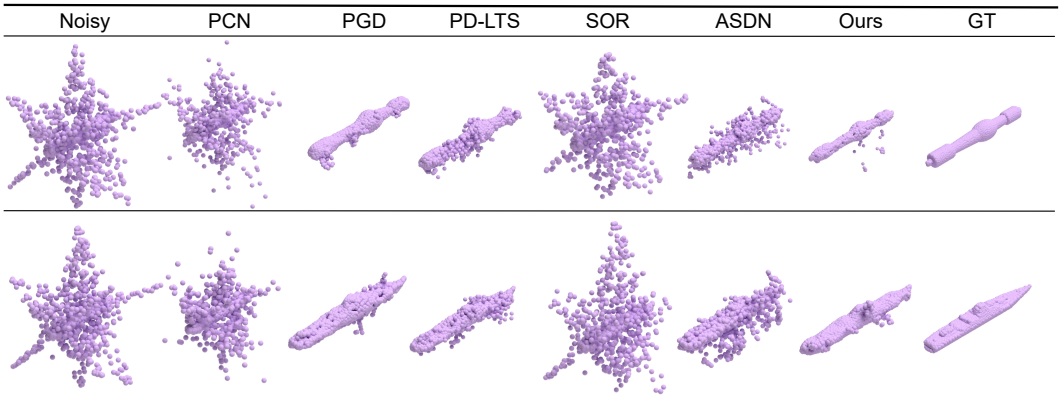


Figure 10: Representative failure cases under High single-photon degradation. Columns show the noisy observation, selected baselines (PCN, PGD, PD-LTS, SOR, and ASDN), PRISM, and the clean ground truth (GT). PRISM recovers the dominant object geometry and removes most diffuse corruption, while two localized limitations remain visible in the shown examples: small residual points can persist near or on the reconstructed surface, and weakly supported fine structures can be smoothed or omitted.