PROTOTYPE-GUIDED DIFFERENTIABLE RANGE OPTIMIZATION FOR ROTATION-ROBUST GEOMETRIC RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Geometric reconstruction seeks to recover complete foundational topologies from partial 3D observations, a task significantly complicated by unknown object orientations. Conventional approaches often rely on canonically aligned training data and process Cartesian coordinates in a pose-dependent manner. Furthermore, severe occlusion restricts the structural evidence required to infer unobserved regions, while errors in rotation-invariant representations frequently do not translate proportionally into Cartesian space. To address these limitations, we propose PA-RangeLM (Prototype-Assisted Range Levenberg–Marquardt), a two-stage framework tailored for rotation-robust geometric reconstruction. To explicitly compensate for missing structures, our Prototype Auxiliary Branch introduces sparse, complete-shape priors during training without modifying the inference architecture. To address coordinate mapping discrepancies, RangeLM integrates Bounded Range Correction and Uncertainty-drived Weighting into a differentiable coordinate recovery process. By encoding partial inputs as distances to local anchors and optimizing coordinate recovery end to end, the framework operates in the observed frame without canonicalization or explicit pose estimation. Empirical evaluations under random-rotation protocols demonstrate that PA-RangeLM consistently outperforms established rotation-robust baselines in standard coordinate metrics. Code and supplementary details are provided for review. Project page: `https://pa-rangelm.github.io/`.

1 INTRODUCTION

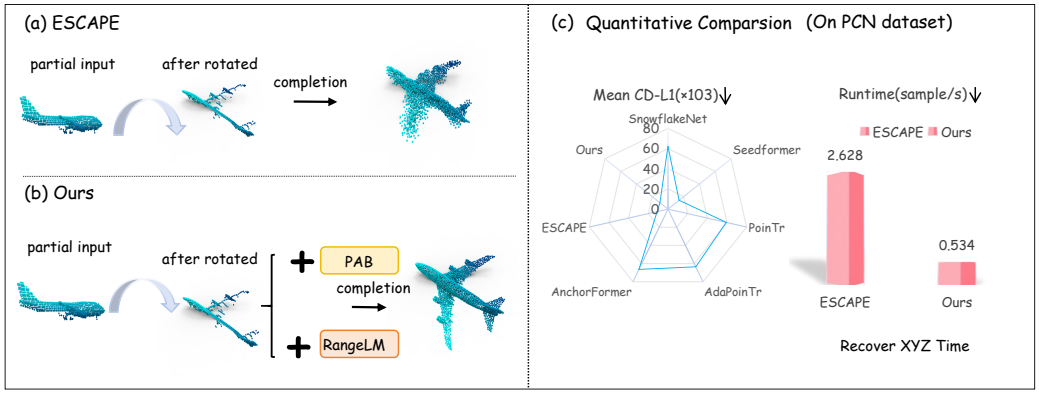

Figure 1: PA-RangeLM completes randomly rotated point clouds in the observed frame. Prototype supervision and differentiable range recovery avoid a separate canonicalization module.

Point clouds directly represent 3D geometry in robotic perception, autonomous systems, and digital reconstruction. Depth cameras and laser scanners, however, observe only partial surfaces because

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.

of occlusion, restricted viewpoints, and measurement noise. Point cloud completion must therefore infer plausible geometry that is absent from the observation. Existing methods learn shape priors through coarse-to-fine generation or proxy-based set translation (Yuan et al., 2018; Yu et al., 2021). Structural tokens, adaptive queries, and regional modulation further improve local geometry (Xiang et al., 2021; Zhou et al., 2022; Yu et al., 2023b; Chen et al., 2023). These methods nevertheless process Cartesian features and generally assume canonical object poses. This assumption is restrictive in real scans, where an object may arrive under an arbitrary orientation and its visible fragment may not determine a unique reference frame. Unoriented inputs consequently require external alignment. Errors in that alignment propagate to the completion network and can be amplified when the observed region is small.

Rotation-robust completion is commonly approached through augmentation, canonicalization, or equivariant architectures (Deng et al., 2021). ESCAPE instead predicts distances to input-derived anchors, avoiding Cartesian canonicalization (Bekci et al., 2025). This representation is robust to joint rotations of the points and anchors, but a partial observation can still support several plausible complete shapes. Moreover, one encoder–decoder path must learn both global topology and dense local geometry. Our first insight is to separate structural supervision from dense generation. A training-only Prototype Auxiliary Branch predicts a sparse complete-shape representation and supervises the shared features without entering the inference path. Learnable queries receive category-level context from the global feature and sample-specific evidence from cross-attention to encoder memory. Their target is sampled from the complete shape before conversion to anchor distances, so the auxiliary task emphasizes spatial coverage rather than duplicating dense point generation. The branch is discarded after Stage I.

The distance-to-coordinate interface creates a second limitation. ESCAPE trains in anchor-distance space but recovers Cartesian coordinates with a non-differentiable post-processing solver (Bekci et al., 2025). Distance errors can therefore produce disproportionate Cartesian errors, especially for ill-conditioned anchor layouts. Our second insight is to train coordinate recovery with the predictor. Inspired by differentiable nonlinear optimization (Muhle et al., 2023; Weng et al., 2026; Höllein et al., 2025), RangeLM learns Bounded Range Correction and Uncertainty-drived Weighting within DLM-CR. The correction absorbs bounded systematic range bias, while the weights reduce the influence of unreliable point–anchor constraints. The resulting Cartesian loss directly supervises the dense prediction path. This design aligns the learned representation with the coordinate-space metric used at evaluation rather than treating multilateration as an isolated final step.

We combine these ideas in **PA-RangeLM**. Stage I uses 128 learnable queries and encoder features to predict a sparse complete-shape prototype in anchor-distance space. Stage II freezes the representation backbone and trains the dense decoder with RangeLM and DLM-CR. Early solver iterations run without gradient tracking, and the final iterations remain differentiable, limiting memory while retaining Cartesian supervision. At test time, the vectorized solver recovers the full dense output in the observed frame. We evaluate this pipeline under the random-rotation protocol in Section 4.1. Because finite numerical updates and implementation details are not strictly equivariant, we claim empirical rotation robustness rather than exact $\mathrm{SO}(3)$-equivariance.

In summary, the principal contributions of this work are threefold:

- A training-only Prototype Auxiliary Branch supplies sparse complete-shape supervision without adding inference-time computation, separating structural learning from dense generation.
- RangeLM combines Bounded Range Correction and Uncertainty-drived Weighting with DLM-CR to align training with Cartesian evaluation while retaining anchor-distance prediction.
- A batched DLM-CR implementation supports end-to-end training and reduces coordinate-recovery time relative to the sequential baseline.

2 Related Work

2.1 Learning Representations from Point Sets

Point cloud networks must process unordered and irregular spatial samples. PointNet uses shared multilayer perceptrons and symmetric pooling (Charles et al., 2017), whereas PointNet++ hierarchically aggregates local features (Qi et al., 2017). For completion, PoinTr formulates set-to-set translation over point proxies (Yu et al., 2021). AdaPoinTr extends this formulation with adaptive queries and denoising objectives (Yu et al., 2023b). These models learn strong shape representations, but their Cartesian features remain pose-dependent.

2.2 Rotation-Robust 3D Representation Learning

Classical rotation-invariant descriptors encode point-pair relations or sampled surface statistics (Drost et al., 2010; Rusu et al., 2009; Osada et al., 2002). Their fixed summaries, however, provide limited capacity for category-level completion. Learned descriptors use autoencoded local geometry, cylindrical representations, or icosahedral groups (Deng et al., 2018; Ao et al., 2021; Wang et al., 2022). Recent Transformers combine invariant local cues with global context for registration (Yu et al., 2023a; 2024).

Rotation-robust completion instead canonicalizes partial inputs (Sajnani et al., 2022) or uses equivariant vector features (Deng et al., 2021; Wu & Miao, 2022; Sen et al., 2023). ESCAPE predicts in a rotation-robust anchor-distance space (Bekci et al., 2025). PA-RangeLM retains this representation while adding complete-shape supervision and differentiable coordinate recovery.

2.3 Point Cloud Completion and Geometric Recovery

Completion decoders use coarse-to-fine generation, grid deformation, trees, or point-grid hybrids (Yuan et al., 2018; Yang et al., 2018; Tchapmi et al., 2019; Xie et al., 2020). Transformer variants refine proxy tokens through splitting, patch seeds, or regional modulation (Xiang et al., 2021; Zhou et al., 2022; Chen et al., 2023). These designs couple structural reasoning with the dense prediction path. Our prototype branch instead provides structural supervision only during training. ShapeFormer and PaCo model probabilistic implicit codes or planar primitives (Yan et al., 2022; Chen et al., 2025). We retain deterministic dense generation and focus on pose robustness.

Recovering coordinates from invariant ranges requires nonlinear least squares. Differentiable solvers have supported uncertainty-aware pose estimation, range correction, and parallel 3D optimization (Moré, 1978; Muhle et al., 2023; Weng et al., 2026; Höllein et al., 2025). RangeLM applies this strategy to point cloud completion by training the distance-to-coordinate interface jointly with the dense predictor.

3 Method

3.1 Problem Formulation and Framework Overview

Let $\mathcal{P} \subset \mathbb{R}^3$ and $\mathcal{X}^* \subset \mathbb{R}^3$ denote a partial input and its complete target. Following ESCAPE (Bekci et al., 2025), a curvature-aware selector extracts anchors $\mathcal{A} = \{\mathbf{a}_k\}_{k=1}^K$ from $\mathcal{P}$. We encode a point $\mathbf{x}$ by its anchor distances:

$$\phi_{\mathcal{A}}(\mathbf{x}) = [\|\mathbf{x} - \mathbf{a}_1\|_2, \ldots, \|\mathbf{x} - \mathbf{a}_K\|_2]\,. \tag{1}$$

For a point set $\mathcal{Y}$, $\Phi_{\mathcal{A}}(\mathcal{Y})$ stacks these vectors. A rigid transform $\mathcal{T}(\mathbf{x}) = \mathbf{R}\mathbf{x} + \mathbf{t}$ preserves the encoding when applied jointly to points and anchors:

$$\phi_{\mathcal{T}(\mathcal{A})}(\mathcal{T}(\mathbf{x})) = \phi_{\mathcal{A}}(\mathbf{x}). \tag{2}$$

The network predicts a complete set in this invariant space and recovers XYZ coordinates in the observed frame. We evaluate the finite-iteration solver's rotation robustness empirically rather than claim exact solver equivariance. Stage I trains a Prototype Auxiliary Branch to supervise complete-shape structure without entering the dense inference path. Stage II freezes the backbone and prototype branch, then trains the dense decoder and RangeLM heads through Differentiable Levenberg–Marquardt Coordinate Recovery (DLM-CR).

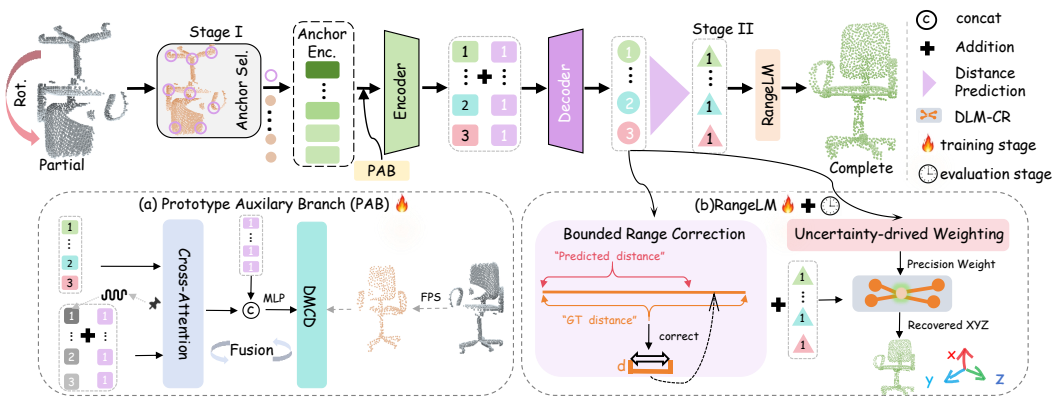

Figure 2: PA-RangeLM architecture. Stage I learns anchor-distance features with training-only prototype supervision. Stage II applies Bounded Range Correction and Uncertainty-drived Weighting before DLM-CR, enabling direct XYZ supervision.

3.2 Rotation-Invariant Distance Backbone

To rigorously isolate the contributions of our proposed modules, we adopt the anchor selection strategy and distance-space Transformer backbone from ESCAPE (Bekci et al., 2025). This ensures all evaluations occur within the same rotation-robust representation space. Formally, the incomplete input is encoded as $\mathbf{D}_{\text{in}} = \Phi_{\mathcal{A}}(\mathcal{P}) \in \mathbb{R}^{N\times K}$, and the complete target is defined as $\mathbf{D}^* = \Phi_{\mathcal{A}}(\mathcal{X}^*) \in \mathbb{R}^{M\times K}$. A graph-based encoder processes this input to yield a memory matrix $\mathbf{H} \in \mathbb{R}^{N_e\times C}$ and a global feature vector $\mathbf{g} \in \mathbb{R}^{C_g}$. The Transformer decoder then processes these representations to output a coarse distance set $\widehat{\mathbf{D}}_{\text{coarse}}$ alongside N_q decoding queries. Crucially, rather than directly regressing Cartesian XYZ coordinates, a dense prediction head expands these queries into the base distance set $\widehat{\mathbf{D}}_{\text{base}} \in \mathbb{R}^{M\times K}$, thereby preserving the invariant coordinate-free formulation prior to the recovery stage.

We use deterministic farthest point sampling to select K dispersed centers, then choose the maximum-curvature point within radius 0.075 of each center; curvature is estimated from the normal covariance of 16 neighbors. This selector depends on relative geometry rather than a fixed global frame.

We compare unordered anchor-distance sets with Distance Matrix Chamfer Distance (DMCD):

$$\text{DMCD}(\mathbf{U}_D, \mathbf{V}_D) = \frac{1}{n}\sum_{i=1}^{n} \min_{1\le j\le m} \|\mathbf{u}_i^D - \mathbf{v}_j^D\|_1 + \frac{1}{m}\sum_{j=1}^{m} \min_{1\le i\le n} \|\mathbf{v}_j^D - \mathbf{u}_i^D\|_1. \tag{3}$$

Here $\mathbf{U}_D$ and $\mathbf{V}_D$ contain n and m distance vectors. DMCD supervises the coarse and dense predictions; we also retain ESCAPE's distance-space denoising task.

3.3 Prototype-Assisted Complete-Shape Supervision

Under severe occlusion, a single partial observation is often compatible with multiple plausible geometries, rendering the reconstruction inherently ill-posed. The standard ESCAPE backbone struggles to resolve this ambiguity because its shared encoder representations lack explicit supervision from a complete global prior. To explicitly break this topological ambiguity, we introduce a training-only Prototype Auxiliary Branch designed to predict a sparse, complete-shape prototype directly from the encoded features.

Formally, let $\mathbf{Q}_{\text{proto}} \in \mathbb{R}^{N_p\times C}$ denote a set of N_p learnable prototype queries. To explicitly inject global context, we first add a projected global feature $f_g(\mathbf{g})$ to each query. The modulated tokens

then attend to the encoder memory $\mathbf{H}$ via cross-attention to aggregate structure-aware features:

$$\begin{cases} \widetilde{\mathbf{Q}}_{\text{proto}} = \mathbf{Q}_{\text{proto}} + f_g(\mathbf{g}), \\ \mathbf{Z}_{\text{proto}} = \widetilde{\mathbf{Q}}_{\text{proto}} + \text{MHA}\Big(\text{LN}(\widetilde{\mathbf{Q}}_{\text{proto}}), \mathbf{H}, \mathbf{H}\Big). \end{cases} \tag{4}$$

Here LN and MHA denote layer normalization and multi-head attention. The enriched tokens are concatenated with the global feature and mapped to anchor-distance space:

$$\widehat{\mathbf{D}}_{\text{proto}} = f_{\text{proto}}([\mathbf{Z}_{\text{proto}} \,\|\, \mathbf{g}]) \in \mathbb{R}^{N_p \times K}. \tag{5}$$

To construct the supervision target, we apply farthest point sampling (FPS) in the Cartesian space of the complete shape, subsequently encoding these sampled points via the input anchors. Formally, $\mathcal{X}^*_{\text{proto}} = \text{FPS}(\mathcal{X}^*, N_p)$ and $\mathbf{D}^*_{\text{proto}} = \Phi_{\mathcal{A}}(\mathcal{X}^*_{\text{proto}})$. The prototype auxiliary loss is thus formulated as:

$$\mathcal{L}_{\text{proto}} = \text{DMCD}\left(\widehat{\mathbf{D}}_{\text{proto}}, \mathbf{D}^*_{\text{proto}}\right). \tag{6}$$

Sampling in Cartesian space before distance conversion preserves spatial coverage. The prototype is never fed to the main decoder, so it supervises the shared encoder in Stage I without inference overhead.

3.4 RangeLM: Bounded Range Correction and Uncertainty-drived Weighting

Relying exclusively on distance-space supervision does not guarantee accurate Cartesian reconstruction following multilateration. In practice, even minor, correlated errors in the predicted ranges can amplify into substantial geometric deviations in XYZ space, particularly when the local anchor configuration is ill-conditioned. Drawing inspiration from recent breakthroughs in differentiable nonlinear least-squares optimization (Muhle et al., 2023; Weng et al., 2026), our RangeLM module addresses this fundamental bottleneck by transforming geometric recovery from an isolated, post-hoc evaluation step into a fully trainable, end-to-end interface.

Bounded Range Correction and Uncertainty-drived Weighting Formally, let $\mathbf{F} \in \mathbb{R}^{N_q \times C_r}$ denote the shared reconstruction features derived from the dense prediction head. To formulate the differentiable recovery, the Bounded Range Correction and Uncertainty-drived Weighting heads project these features into matrices $f_{\text{corr}}(\mathbf{F}), f_u(\mathbf{F}) \in \mathbb{R}^{M \times K}$. Denoting the individual base distance elements as $\widehat{d}^{\text{base}}_{ik}$, we define Bounded Range Correction as:

$$\begin{cases} \delta_{ik} = \rho \tanh(f_{\text{corr}}(\mathbf{F})_{ik}), \\ d^{\text{corr}}_{ik} = \max\Big(\varepsilon, \widehat{d}^{\text{base}}_{ik} + \delta_{ik}\Big). \end{cases} \tag{7}$$

The positive scalars ρ and ε bound the correction and impose a range floor. The final correction layer is zero-initialized, so Stage II begins from the base ranges.

Concurrently, the Uncertainty-drived Weighting head predicts a bounded log-variance $s_{ik} = \text{clip}(f_u(\mathbf{F})_{ik}, s_{\min}, s_{\max})$ for each corrected range, parameterized by scalars s_{ik}, $s_{\min}$, and $s_{\max}$. We project these log-variances into normalized precision weights w_{ik}:

$$w_{ik} = w_{\min} + (1 - w_{\min}) K \frac{\exp(-s_{ik})}{\sum_{\ell=1}^{K} \exp(-s_{i\ell})}. \tag{8}$$

The zero-initialized head begins with uniform weights. We interpret them as task-specific relative weights, not calibrated uncertainties.

Differentiable Levenberg–Marquardt coordinate recovery Formally, the corrected ranges and their corresponding weights establish a weighted nonlinear least-squares problem for each predicted point:

$$\widehat{\mathbf{x}}_i = \underset{\mathbf{x} \in \mathbb{R}^3}{\arg\min} \frac{1}{2} \sum_{k=1}^{K} w_{ik} \left(\|\mathbf{x} - \mathbf{a}_k\|_2 - d^{\text{corr}}_{ik}\right)^2. \tag{9}$$

DLM-CR solves this point-wise objective with adaptive damping and retains gradients through its final training iterations; Appendix B.2 gives the update and safeguards.

To limit memory, early LM iterations run without gradient tracking and only the final refinements are differentiated. Training recovers a sampled subset of points, whereas evaluation recovers the full output.

3.5 Two-Stage Optimization

During Stage I, we jointly optimize the distance-aware backbone and the Prototype Auxiliary Branch. Letting $\mathcal{L}_{\text{denoise}}$ represent the anchor-space denoising objective, we formally define the coarse and dense structural losses as $\mathcal{L}_{\text{coarse}} = \text{DMCD}(\widehat{\mathbf{D}}_{\text{coarse}}, \mathbf{D}^*)$ and $\mathcal{L}^{\text{I}}_{\text{dense}} = \text{DMCD}(\widehat{\mathbf{D}}_{\text{base}}, \mathbf{D}^*)$, respectively. The comprehensive Stage-I loss function is thus formulated as:

$$\mathcal{L}_{\text{I}} = \lambda_{\text{denoise}}\mathcal{L}_{\text{denoise}} + \mathcal{L}_{\text{coarse}} + \mathcal{L}^{\text{I}}_{\text{dense}} + \lambda_{\text{proto}}\mathcal{L}_{\text{proto}}. \tag{10}$$

We use $\lambda_{\text{denoise}} = 0.5$ and $\lambda_{\text{proto}} = 0.1$; no Cartesian recovery is applied in Stage I.

Building upon the optimized representations established in Stage I, Stage II targets the end-to-end differentiable coordinate recovery. To ensure optimization stability, we strictly freeze the representation backbone, the coarse anchor pathway, and the Prototype Auxiliary Branch. Consequently, optimization is directed exclusively toward the Transformer decoding path and the RangeLM heads. We first define the corrected dense-distance loss as $\mathcal{L}^{\text{II}}_{\text{dense}} = \text{DMCD}(\mathbf{D}^{\text{corr}}, \mathbf{D}^*)$. To formulate the Cartesian supervision, let M_s denote the number of points stochastically sampled for the DLM-CR solver, with $\widehat{\mathcal{X}}_s$ and $\mathcal{X}^*_s$ representing the recovered XYZ predictions and their corresponding ground truth. For any two Cartesian point sets $\mathcal{U}_X = \{\mathbf{u}^X_i\}_{i=1}^n$ and $\mathcal{V}_X = \{\mathbf{v}^X_j\}_{j=1}^m$, the coordinate-space Chamfer Distance is defined as:

$$\text{CD}_{\text{XYZ}}(\mathcal{U}_X, \mathcal{V}_X) = \frac{1}{n}\sum_{i=1}^{n}\min_{1\le j\le m}\|\mathbf{u}^X_i - \mathbf{v}^X_j\|_1 + \frac{1}{m}\sum_{j=1}^{m}\min_{1\le i\le n}\|\mathbf{v}^X_j - \mathbf{u}^X_i\|_1. \tag{11}$$

Alongside the Cartesian-space loss $\mathcal{L}_{\text{xyz}}$, we introduce two RangeLM regularizers: $\mathcal{L}_{\text{corr}}$ to penalize extreme range deviations, and $\mathcal{L}_u$ to constrain the log-variance magnitudes:

$$\begin{cases} \mathcal{L}_{\text{xyz}} = \text{CD}_{\text{XYZ}}\left(\widehat{\mathcal{X}}_s, \mathcal{X}^*_s\right), \\ \mathcal{L}_{\text{corr}} = \dfrac{1}{MK}\displaystyle\sum_{i=1}^{M}\sum_{k=1}^{K}\left|d^{\text{corr}}_{ik} - \widehat{d}^{\text{base}}_{ik}\right|, \\ \mathcal{L}_u = \dfrac{1}{M_sK}\displaystyle\sum_{i=1}^{M_s}\sum_{k=1}^{K} s^2_{ik}. \end{cases} \tag{12}$$

The effective Stage-II objective for the active parameter set thus becomes:

$$\mathcal{L}_{\text{II}} = \lambda_{\text{denoise}}\mathcal{L}_{\text{denoise}} + \mathcal{L}^{\text{II}}_{\text{dense}} + \lambda_{\text{xyz}}\mathcal{L}_{\text{xyz}} + \lambda_{\text{corr}}\mathcal{L}_{\text{corr}} + \lambda_u\mathcal{L}_u. \tag{13}$$

We set $\lambda_{\text{denoise}} = 0.5$, $\lambda_{\text{xyz}} = 1$, $\lambda_{\text{corr}} = 0.01$, and $\lambda_u = 10^{-4}$. Coarse and prototype losses are logged but detached in Stage II. Inference applies the dense decoder, Bounded Range Correction, Uncertainty-drived Weighting, and DLM-CR.

4 Experiments

We evaluate PA-RangeLM on randomly rotated PCN shapes, transfer the PCN-trained models to MVP and KITTI, and analyze the two proposed modules and solver runtime.

4.1 Experimental Setup

Datasets and evaluation protocols PCN (Yuan et al., 2018) contains eight ShapeNet-derived categories. We use its official 28,974/800/1,200 train/validation/test split. Models are trained on canonical poses and tested after independently sampling three Euler angles from $[-180°, 180°]$ with seed 2026. Each rotation is applied jointly to the input, target, and anchors. Non-equivariant baselines use ConDor canonicalization (Sajnani et al., 2022; Bekci et al., 2025), whereas ESCAPE and PA-RangeLM operate directly on the observed pose.

For cross-dataset evaluation, the PCN-trained models are applied without fine-tuning to 31,200 MVP pairs from the eight shared categories (Pan et al., 2021). MVP uses the same rotation protocol. We also evaluate the PCN car model on real partial KITTI scans (Geiger et al., 2012). Because KITTI lacks complete ground truth, Table 3 states the protocol difference between our evaluation and the quoted baselines.

Evaluation metrics We evaluate PCN and MVP using CD-$L_1 \downarrow (\times 10^3)$. KITTI has no complete ground truth, so we report Fidelity $\downarrow$ and Minimum Matching Distance (MMD $\downarrow$), both scaled by 10^3; Appendix B.3 gives their definitions.

Implementation details Training uses PyTorch on one NVIDIA RTX 3090 with batch size 16, eight anchors, 512 decoder queries, and 128 prototype queries. Stage I runs for 300 epochs and Stage II for 50 epochs. Training samples 4,096 points and uses five detached warm-up iterations followed by three differentiable LM iterations. Evaluation recovers all 16,384 points with at most 80 iterations. Appendix B.1 reports the optimizer, schedules, solver bounds, checkpoint selection, and randomness controls.

4.2 Completion under the Evaluated Random-Rotation Protocol on PCN

Quantitative PCN results under the random-rotation protocol in Section 4.1 are presented in Table 1. While conventional non-equivariant baselines fundamentally rely on pre-processing via ConDor canonicalization to mitigate pose variations, they still exhibit severe vulnerability to unconstrained rotations, with most suffering catastrophic performance drops (e.g., AnchorFormer at 66.40). In stark contrast, both ESCAPE and our proposed PA-RangeLM natively process the observed unconstrained poses across the full 1,200-sample test split. Under these rigorous conditions, PA-RangeLM achieves a superior average Cartesian CD-L_1 of 10.20. This decisively outperforms not only the best-performing canonicalized baseline (SeedFormer at 14.07) but also the strong ESCAPE backbone (11.01). Crucially, this performance margin is consistent across all eight distinct object categories, including structurally complex classes such as Sofa (13.90 versus 14.45). While we rigorously confine our formal empirical claims to this established test split and the evaluated unconstrained $\mathrm{SO}(3)$ protocol, these consistent gains conclusively validate the advantage of our differentiable geometric recovery. Furthermore, the visual comparisons in Figure 3 qualitatively corroborate these metrics, demonstrating that PA-RangeLM recovers higher-fidelity geometry and resolves structural ambiguities even under extreme pose perturbations.

Table 1: PCN completion under random $\mathrm{SO}(3)$ rotations (seed 2026), measured by CD-$L_1 \downarrow (\times 10^3)$. ESCAPE-Paper is from Bekci et al. (2025); ESCAPE-Reprod. is our reproduction.

Category	Snowflake Net (ICCV'21)	Seed Former (ECCV'22)	PoinTr (ICCV'21)	Ada PoinTr (TPAMI'23)	Anchor Former (CVPR'23)	ESCAPE-Paper (CVPR'25)	ESCAPE-Reprod. (CVPR'25)	Ours
Plane	70.49	16.39	63.81	65.34	66.15	8.60	8.98	**7.58**
Cabin	57.42	11.72	58.04	65.11	67.59	13.62	13.89	**13.04**
Car	56.23	11.40	54.14	56.17	60.81	10.43	10.77	**10.07**
Chair	59.31	13.02	60.07	63.73	67.88	10.71	11.15	**10.36**
Lamp	69.61	17.68	64.74	68.33	68.39	8.14	8.61	**7.93**
Sofa	63.22	15.35	61.70	66.69	70.06	**13.86**	14.45	13.90
Table	63.35	14.79	62.94	72.52	76.64	9.23	9.87	**9.09**
Boat	58.40	12.21	49.07	50.42	53.68	10.00	10.36	**9.60**
Average	62.26	14.07	59.31	63.54	66.40	10.58	11.01	**10.20**

The category breakdown provides a direct comparison with the reproduced ESCAPE checkpoint. PA-RangeLM lowers CD-L_1 in every category, with absolute reductions ranging from 0.55 for sofa to 1.40 for plane. Relative to the values reported in the ESCAPE paper, PA-RangeLM is lower in seven categories; sofa is nearly tied at 13.90 versus 13.86. We therefore use ESCAPE-Reprod. as the within-implementation comparison and retain ESCAPE-Paper as reported context.

4.3 Cross-Dataset Evaluation on MVP

Table 2 evaluates the PCN-trained models on MVP without fine-tuning. PA-RangeLM lowers the eight-category average from 12.06 to 11.38 (5.7%) and improves all shared categories under the evaluated rotation protocol.

Table 2: PCN-to-MVP transfer on eight shared categories under random $\mathrm{SO}(3)$ rotations, measured by CD-$L_1 \downarrow (\times 10^3)$.

Method	Plane	Cabin	Car	Chair	Lamp	Sofa	Table	Watercraft	Average
ESCAPE (CVPR'25)	7.53	15.88	12.35	12.60	10.20	14.34	11.90	11.66	12.06
Ours	6.63	15.32	11.70	12.11	9.43	13.46	11.19	11.16	11.38

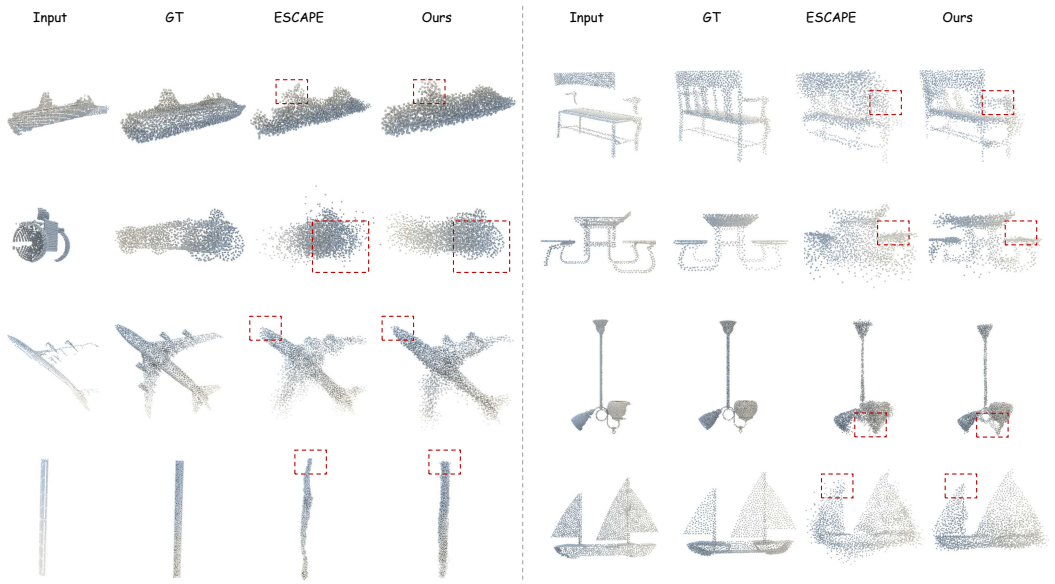

Figure 3: Qualitative completion results. The left block shows PCN visualizations, and the right block shows MVP visualizations from the categories shared with PCN.

4.4 Completion of KITTI Cars

Table 3 evaluates the PCN-trained car model on real KITTI scans. Quoted baselines use random single-axis rotations; PA-RangeLM uses normalized scans, so the values document behavior rather than a controlled ranking.

4.5 Runtime Profiling of the Differentiable LM Solver

On the 1,200-sample PCN test set, DLM-CR reduces recovery time from 2.478 to 0.384 s/sample and total time from 2.628 to 0.534 s/sample (Table 4).

Table 3: KITTI car completion, measured by Fidelity $\downarrow$ and MMD $\downarrow (\times 10^3)$.

Metric	PoinTr (ICCV'21)	Snowflake Net (ICCV'21)	ESCAPE (CVPR'25)	Ours
Fidelity $\downarrow$	**0.00**	1.77	1.81	1.59
MMD $\downarrow$	6.15	16.08	**5.93**	8.72

Table 4: Mean runtime $\downarrow$ (s/sample) on the 1,200-sample rotated PCN test set.

Backend	Network $\downarrow$	Recovery $\downarrow$	Total $\downarrow$
ESCAPE sequential (CVPR'25)	**0.150**	2.478	2.628
DLM-CR	**0.150**	**0.384**	**0.534**

4.6 Ablation Studies

We ablate both proposed modules while fixing the backbone, output resolution, and rotation protocol.

Prototype Auxiliary Branch Table 5 starts from the backbone without the auxiliary branch (10.54). Global conditioning alone gives 10.49, whereas memory cross-attention alone gives 10.36. Combining both inputs yields 10.20, showing that global category context and input-specific encoder memory are complementary.

Table 5: Prototype Auxiliary Branch ablation on rotated PCN, measured by CD-$L_1 \downarrow$ ($\times 10^3$).

Variant	PCN CD-$L_1 \downarrow$
Base	10.54
+ Global conditioning	10.49
+ Memory cross-attention	10.36
Full (global + memory)	**10.20**

Bounded Range Correction and Uncertainty-drived Weighting With neither RangeLM component, the Stage-II baseline obtains 10.48 (Table 6). Bounded Range Correction lowers this to 10.36, and adding Uncertainty-drived Weighting reaches 10.20. Thus both the range adjustment and its point-anchor weighting contribute under the same Stage-I initialization.

Table 6: RangeLM ablation on rotated PCN using one Stage-I checkpoint; CD-$L_1 \downarrow$ ($\times 10^3$).

Variant	PCN CD-$L_1 \downarrow$
Base	10.48
+ Bounded Range Correction	10.36
+ Uncertainty-drived Weighting (Full)	**10.20**

The two ablations address different optimization stages. Table 5 changes the supervision used to learn the Stage-I representation, whereas Table 6 retains one Stage-I checkpoint and modifies the Stage-II recovery interface. Their identical 10.20 endpoints do not imply interchangeable mechanisms: the Prototype Auxiliary Branch is absent at inference, while both RangeLM components remain in the deployed path. Together, these results support the selected two-stage configuration under the stated checkpoint and rotation protocol.

Accordingly, the synthetic benchmarks, real-scan transfer, and runtime profile are treated as separate pieces of evidence rather than one aggregate ranking.

These results should be interpreted within the evaluated protocols. The PCN comparison covers the full 1,200-sample test split under sampled $\mathrm{SO}(3)$ rotations. The MVP experiment reuses the PCN-trained checkpoint without fine-tuning, so it measures transfer rather than adaptation to the target dataset. Because the shared subset changes both category coverage and dataset statistics, this experiment does not isolate either source of shift. For KITTI, complete targets are unavailable, and the compared entries use different pose protocols. Its metrics therefore document transfer behavior under the stated settings rather than a controlled ranking against complete ground truth. The component ablations likewise use fixed checkpoints and single reported runs. They isolate the tested design choices but do not estimate run-to-run variability or all joint removals. Overall, the experiments support the reported configurations under their stated rotation, dataset, and checkpoint conditions. Claims about broader robustness require matched real-scan protocols, additional sensors and object families, alternative anchor settings, and repeated trials. Such tests would help distinguish gains from the representation from those tied to the benchmark construction.

5 Conclusion

We presented PA-RangeLM, an end-to-end differentiable framework for point cloud completion under unconstrained poses. By integrating a training-time Prototype Auxiliary Branch, Bounded Range Correction, and Uncertainty-drived Weighting, our architecture elegantly resolves the inherent ambiguity of hallucinating missing geometry and the error amplification intrinsic to Cartesian recovery. Extensive evaluations demonstrate that PA-RangeLM decisively outperforms strong baselines under rigorous random-rotation protocols on PCN, while exhibiting robust zero-shot transferability on MVP. Furthermore, our fully vectorized geometric solver massively accelerates inference. By consciously trading strict mathematical equivariance for numerical stability and maximal hardware throughput, PA-RangeLM establishes a formidable and highly practical paradigm for real-world 3D vision applications where object orientation is fundamentally unconstrained. Future work will evaluate matched real-scan protocols across additional sensor settings.

AI USE STATEMENT

In this work, generative AI tools were used to discuss and refine research methodology, assist with software implementation and debugging, search and summarize literature, and draft and edit parts of the manuscript. The authors checked AI-assisted technical descriptions against the cited papers, source code, and experiment logs, and manually reviewed the resulting text and code. The authors take responsibility for the final content of this work, including claims and artifacts produced with the aid of generative AI.

REFERENCES

Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11753–11762, June 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Ao_SpinNet_Learning_a_General_Surface_Descriptor_for_3D_Point_Cloud_CVPR_2021_paper.html.

Burak Bekci, Nassir Navab, Federico Tombari, and Mahdi Saleh. Escape: Equivariant shape completion via anchor point encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6480–6489, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Bekci_ESCAPE_Equivariant_Shape_Completion_via_Anchor_Point_Encoding_CVPR_2025_paper.html.

R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation . In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85. IEEE Computer Society, Jul 2017. URL https://doi.org/10.1109/CVPR.2017.16.

Zhikai Chen, Fuchen Long, Zhaofan Qiu, Ting Yao, Wengang Zhou, Jiebo Luo, and Tao Mei. Anchorformer: Point cloud completion from discriminative nodes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13581–13590, June 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Chen_AnchorFormer_Point_Cloud_Completion_From_Discriminative_Nodes_CVPR_2023_paper.html.

Zhikai Chen, Fuchen Long, Zhaofan Qiu, Ting Yao, Wengang Zhou, Tao Mei, and Jiebo Luo. Parametric point cloud completion for polygonal surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11749–11758, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Chen_Parametric_Point_Cloud_Completion_for_Polygonal_Surface_Reconstruction_CVPR_2025_paper.html.

Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J. Guibas. Vector neurons: A general framework for so(3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12200–12209, October 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Deng_Vector_Neurons_A_General_Framework_for_SO3-Equivariant_Networks_ICCV_2021_paper.html.

Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. URL https://openaccess.thecvf.com/content_ECCV_2018/html/Tolga_Birdal_PPF-FoldNet_Unsupervised_Learning_ECCV_2018_paper.html.

Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 998–1005, 2010. URL https://doi.org/10.1109/CVPR.2010.5540108.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012. URL https://doi.org/10.1109/CVPR.2012.6248074.

Lukas Höllein, Aljaž Božič, Michael Zollhöfer, and Matthias Nießner. 3dgs-lm: Faster gaussian-splatting optimization with levenberg-marquardt. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26740–26750, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Hollein_3DGS-LM_Faster_Gaussian-Splatting_Optimization_with_Levenberg-Marquardt_ICCV_2025_paper.html.

Jorge J. Moré. The levenberg-marquardt algorithm: Implementation and theory. In *Numerical Analysis*, pp. 105–116. Springer Berlin Heidelberg, 1978. ISBN 978-3-540-35972-2. URL https://doi.org/10.1007/BFb0067700.

Dominik Muhle, Lukas Koestler, Krishna Murthy Jatavallabhula, and Daniel Cremers. Learning correspondence uncertainty via differentiable nonlinear least squares. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13102–13112, June 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Muhle_Learning_Correspondence_Uncertainty_via_Differentiable_Nonlinear_Least_Squares_CVPR_2023_paper.html.

Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, October 2002. URL https://pixl.cs.princeton.edu/pubs/Osada_2002_SD/index.php.

Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational relational point completion network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8524–8533, June 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Pan_Variational_Relational_Point_Completion_Network_CVPR_2021_paper.html.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf.

Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation*, pp. 3212–3217, 2009. URL https://doi.org/10.1109/ROBOT.2009.5152473.

Rahul Sajnani, Adrien Poulenard, Jivitesh Jain, Radhika Dua, Leonidas J. Guibas, and Srinath Sridhar. Condor: Self-supervised canonicalization of 3d pose for partial shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16969–16979, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Sajnani_ConDor_Self-Supervised_Canonicalization_of_3D_Pose_for_Partial_Shapes_CVPR_2022_paper.html.

Bipasha Sen, Aditya Agarwal, Gaurav Singh, Brojeshwar B., Srinath Sridhar, and Madhava Krishna. Scarp: 3d shape completion in arbitrary poses for improved grasping. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3838–3845, 2023. URL https://doi.org/10.1109/ICRA48891.2023.10160365.

Lyne P. Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 383–392, 2019. URL https://doi.org/10.1109/CVPR.2019.00047.

Haiping Wang, Yuan Liu, Zhen Dong, and Wenping Wang. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, pp. 1630–1641. Association for Computing Machinery, 2022. ISBN 9781450392037. URL https://doi.org/10.1145/3503161.3548023.

Xu Weng, Keck Voon Ling, Haochen Liu, Bingheng Wang, and Kun Cao. Nerc: Neural ranging correction through differentiable moving horizon location estimation. In *Proceedings of the 2026 ACM/IEEE International Conference on Embedded Artificial Intelligence and Sensing Systems*, SenSys '26, pp. 903–917. Association for Computing Machinery, 2026. ISBN 9798400723094. URL https://doi.org/10.1145/3774906.3800475.

Hang Wu and Yubin Miao. So(3) rotation equivariant point cloud completion using attention-based vector neurons. In *2022 International Conference on 3D Vision (3DV)*, pp. 280–290, 2022. URL https://doi.org/10.1109/3DV57658.2022.00040.

Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. OmniObject3D: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 803–814, 2023. URL https://doi.org/10.1109/CVPR52729.2023.00084.

Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5499–5509, October 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Xiang_SnowflakeNet_Point_Cloud_Completion_by_Snowflake_Point_Deconvolution_With_Skip-Transformer_ICCV_2021_paper.html.

Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *Computer Vision – ECCV 2020*, pp. 365–381. Springer International Publishing, 2020. ISBN 978-3-030-58545-7. URL https://doi.org/10.1007/978-3-030-58545-7_21.

Xingguang Yan, Liqiang Lin, Niloy J. Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Shapeformer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6239–6249, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Yan_ShapeFormer_Transformer-Based_Shape_Completion_via_Sparse_Representation_CVPR_2022_paper.html.

Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation . In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 206–215. IEEE Computer Society, Jun 2018. URL https://doi.org/10.1109/CVPR.2018.00029.

Hao Yu, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, and Slobodan Ilic. Rotation-invariant transformer for point cloud matching. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5384–5393, 2023a. URL https://doi.org/10.1109/CVPR52729.2023.00521.

Hao Yu, Ji Hou, Zheng Qin, Mahdi Saleh, Ivan Shugurov, Kai Wang, Benjamin Busam, and Slobodan Ilic. Riga: Rotation-invariant and globally-aware descriptors for point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3796–3812, 2024. URL https://doi.org/10.1109/TPAMI.2023.3349199.

Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12498–12507, October 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Yu_PoinTr_Diverse_Point_Cloud_Completion_With_Geometry-Aware_Transformers_ICCV_2021_paper.html.

Xumin Yu, Yongming Rao, Ziyi Wang, Jiwen Lu, and Jie Zhou. Adapointr: Diverse point cloud completion with adaptive geometry-aware transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):14114–14130, 2023b. URL https://doi.org/10.1109/TPAMI.2023.3309253.

Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pp. 728–737, 2018. URL https://doi.org/10.1109/3DV.2018.00088.

Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. Seedformer: Patch seeds based point cloud completion with upsample transformer. In *Computer Vision – ECCV 2022*, pp. 416–432. Springer Nature Switzerland, 2022. ISBN 978-3-031-20062-5. URL https://doi.org/10.1007/978-3-031-20062-5_24.

SUPPLEMENTARY MATERIAL

This supplement records the reproduction choices, technical definitions, and additional experiments that could not be included in the main paper.

A REPRODUCTION AND BENCHMARK SELECTION

A.1 ESCAPE REPRODUCTION ON PCN

The ESCAPE paper reports an average PCN CD-$L_1 \times 10^3$ of 10.58 under random rotations in its Table 1 (Bekci et al., 2025). Table 1 instead lists 11.01, which is our reproduction from the authors' public GitHub release rather than a transcription of the published result.

Source and training. We did not change the released model, training, or coordinate-recovery code. The audit traced the evaluated files to commit `39410a820772`: 25 of 26 tracked files were byte-identical, and the only changed file was the PCN configuration containing local dataset paths. We followed the released script's 250-epoch budget and 50-epoch early-stopping patience. Evaluation used `val_best.pth`, saved at epoch 243.

Evaluation and reporting. The evaluation used all 1,200 PCN test samples and independently sampled three rotation angles in $[-180°, 180°]$ with seed 2026. Coordinate recovery used the released point-wise SciPy LM implementation. The resulting mean CD-$L_1 \times 10^3$ was 11.008113, reported as 11.01 in Table 1. We report this reproduced value because it uses the same test split, rotation manifest, and metric implementation as our method.

Relation to the published result. The paper describes Adam with an initial learning rate of 10^{-3}, early stopping after 30 epochs without validation improvement, and a 200-epoch limit (Bekci et al., 2025). The released configuration used here specifies AdamW, an initial learning rate of 10^{-4}, and weight decay of 5×10^{-4}. The paper does not identify the checkpoint or random seed used for its Table 1 result. The available paper, checkpoint, and logs therefore do not isolate a single cause for the difference between 10.58 and 11.01. We label 11.01 as a reproduced result and do not attribute the gap to an implementation error.

A.2 OMNIOBJECT3D REPRODUCTION AND BENCHMARK SELECTION

OmniObject3D (Wu et al., 2023) was our initial choice for the cross-dataset experiment. We trained ESCAPE using its public implementation on PCN and evaluated two checkpoints on 2,800 car views. Neither run matched the reported car-category median of 26.30 (Bekci et al., 2025) under our reconstructed preprocessing protocol (Table 7).

Table 7: ESCAPE reproduction on 2,800 OmniObject3D car views. Values are median per-sample XYZ CD-$L_1 \times 10^3$; lower is better. The published value is included for context; the other rows use our reconstructed preprocessing protocol. Baseline [6] is ESCAPE (Bekci et al., 2025).

Result source	Car	Difference from reported
ESCAPE paper (CVPR'25)	26.3000	–
Our reproduction, `train_last.pth`	48.5328	+22.2328
Our reproduction, `val_best.pth`	49.3778	+23.0778

On a separate 28-view subset, we varied centering, scale normalization, and the camera principal point. The lowest median, 37.3645, required ground-truth bounding-box normalization. This target-dependent operation is unavailable at inference and was excluded from the reported evaluation. None of the tested settings closed the remaining gap. This does not indicate that the published value is incorrect; it shows that our reconstructed protocol did not reproduce it.

We therefore did not use this OmniObject3D reproduction for method comparison, because an unresolved protocol difference could be mistaken for a model difference. We instead used MVP and evaluated both PCN-trained models with the same preprocessing and metric implementation, without MVP training or fine-tuning.

A.3 MVP Dataset Source and Evaluation Subset

The Multi-View Partial (MVP) Point Cloud Dataset was introduced by Pan et al. (Pan et al., 2021). It contains partial scans rendered from 26 camera poses for each 3D CAD model. We obtained the single-view point-cloud completion benchmark from the authors' official dataset page, `https://mvp-dataset.github.io/`. The primary PCN-to-MVP comparison uses the eight categories shared with PCN: airplane, cabinet, car, chair, lamp, sofa, table, and watercraft. It includes all 26 partial views, giving 31,200 partial–complete test pairs. The other eight MVP categories are evaluated in Appendix C.5. As stated in Section 4.1, the PCN-trained checkpoints are evaluated without MVP training or fine-tuning.

B Technical Details and Metric Definitions

B.1 Training and Evaluation Configuration

Each PCN input contains 2,048 points and each dense target contains 16,384 points. The official split contains 28,974 training, 800 validation, and 1,200 test samples. The MVP evaluation uses all 26 partial views for each shape in the eight categories shared with PCN, yielding 31,200 pairs, or 3,900 per category. The PCN-trained checkpoints are applied to MVP and KITTI without fine-tuning.

Both training stages use AdamW on one NVIDIA RTX 3090 with batch size 16, eight curvature-aware anchors, 512 decoder queries, and 128 prototype queries. Stage I runs for 300 epochs with initial learning rate 10^{-4}, weight decay 5×10^{-4}, and multiplier $\max(0.9^{e/21}, 0.02)$ at epoch e. The checkpoint with the lowest validation dense-distance CD-L_1 initializes Stage II. Stage II runs for 50 epochs with initial learning rate 2×10^{-5} and multiplier $\max(0.8^{e/10}, 0.1)$. Its checkpoint is selected by validation Cartesian CD-L_1 after DLM-CR recovery.

RangeLM uses $\rho = 0.05$, $\varepsilon = 10^{-6}$, $s_{\min} = -4$, $s_{\max} = 4$, $w_{\min} = 0.05$, and initial damping 10^{-3}. Training samples 4,096 predicted and target points, performs five detached warm-up iterations, and then performs three differentiable iterations. Validation and testing recover all 16,384 points with at most 80 iterations. Seed 2026 initializes Python, NumPy, PyTorch, and CUDA. Strict cuDNN determinism is disabled for throughput; no optimization instability was observed in the reported runs.

B.2 DLM-CR Implementation Details

For each predicted point, DLM-CR solves the weighted least-squares problem in Eq. (9) with the notation used in Section 3.4. The point estimate $\mathbf{x}_i \in \mathbb{R}^3$ is a three-dimensional vector, each residual $r_{ik} \in \mathbb{R}$ is a scalar, and the update $\Delta\mathbf{x}_i \in \mathbb{R}^3$ is a vector. The damping coefficient $\mu_i \in \mathbb{R}_{>0}$ is a scalar maintained independently for each point. Specifically, $r_{ik} = \|\mathbf{x}_i - \mathbf{a}_k\|_2 - d_{ik}^{\mathrm{corr}}$, $\mathbf{J}_{ik} = ((\mathbf{x}_i - \mathbf{a}_k)/\max(\|\mathbf{x}_i - \mathbf{a}_k\|_2, \epsilon_{\mathrm{num}}))^{\mathsf{T}}$, and $\mathbf{W}_i = \mathrm{diag}(w_{i1}, \ldots, w_{iK})$.

For reproducibility, the implementation forms $\mathbf{H}_i = \mathbf{J}_i^{\mathsf{T}}\mathbf{W}_i\mathbf{J}_i$ and $\mathbf{g}_i = \mathbf{J}_i^{\mathsf{T}}\mathbf{W}_i\mathbf{r}_i$ and solves

$$[\mathbf{H}_i + \mu_i \,\mathrm{diag}(\max(\mathrm{diag}(\mathbf{H}_i), \epsilon_{\mathrm{num}})) + \epsilon_{\mathrm{num}}\mathbf{I}]\,\Delta\mathbf{x}_i = -\mathbf{g}_i. \tag{14}$$

Here, $\mathbf{I} \in \mathbb{R}^{3\times3}$ is the identity matrix, $\mathrm{diag}(\mathbf{H}_i)$ extracts the diagonal of $\mathbf{H}_i$, and the maximum is applied elementwise. This diagonal scaling is expressed in the current Cartesian coordinates and is not rotation-covariant in general. Accordingly, we make no claim that a finite sequence of Eq. (14) is exactly $\mathrm{SO}(3)$-equivariant. Our rotation claims are limited to empirical performance under the evaluated protocol; Eq. (2) applies to the range objective, not to every numerical update.

Unless an initial estimate is explicitly supplied by the truncated training solve, all points are initialized at the origin, $\mathbf{x}_i^{(0)} = \mathbf{0} \in \mathbb{R}^3$. Training first performs five no-gradient LM iterations from this initialization and then performs three differentiable iterations initialized by the detached result of the warm-up. Validation and testing start from the origin and use the full coordinate-recovery budget of at most 80 iterations. Thus, both modes share the same origin initialization; only the training-time truncated backward pass reuses the warm-up estimate.

At each iteration, the candidate $\mathbf{x}_i + \Delta\mathbf{x}_i$ is accepted only if it is finite and strictly decreases the current weighted objective. Accepted updates multiply the scalar damping by 0.3, while rejected

Table 8: PA-RangeLM performance on the standard-pose and independently rotated PCN test sets. The two evaluations use the same checkpoint and 150 samples per category. Values are XYZ CD-$L_1 \times 10^3$; lower is better. The final column is the standard-pose value minus the rotated value.

Category	Standard pose	Independent rotations	Difference
Airplane	7.6482	7.5871	0.0611
Cabinet	12.9892	13.0399	−0.0507
Car	10.0863	10.0670	0.0193
Chair	10.3507	10.3649	−0.0142
Lamp	8.3436	7.9296	0.4140
Sofa	13.9387	13.9008	0.0379
Table	9.0990	9.0871	0.0119
Watercraft	9.6278	9.6071	0.0207
Average	10.2604	10.1979	0.0625

updates multiply it by 10; the coefficient is clipped to the configured positive range $[10^{-12}, 10^{12}]$. The initial damping is 10^{-3}. All reported runs use single-precision recovery. Therefore the effective numerical safeguard is $\epsilon_{\text{num}} = 10^{-8}$, and the effective relative step, objective, and gradient thresholds are 10^{-6}, 10^{-8}, and 10^{-6}, respectively. The batch terminates at the iteration limit, when all training points meet a stopping condition, or, during validation and testing, when the converged fraction reaches 0.999.

To avoid numerical failures for degenerate anchor configurations, the distance Jacobian uses the same positive safeguard, and non-finite steps or candidate objectives are rejected. This numerical ϵ_{num} is distinct from the corrected-range floor $\varepsilon = 10^{-6}$ in Eq. (7).

B.3 KITTI Metric Definitions

KITTI does not provide complete ground-truth geometry for its partial LiDAR scans. Let $\mathcal{P}_i$ be the observed scan for sample i, $\widehat{\mathcal{X}}_i$ its predicted completion, and $\mathcal{R}$ the set of complete PCN car shapes used as references. The one-sided Fidelity metric is

$$\text{Fidelity}_i = \frac{1}{|\mathcal{P}_i|} \sum_{\mathbf{p} \in \mathcal{P}_i} \min_{\widehat{\mathbf{x}} \in \widehat{\mathcal{X}}_i} \|\mathbf{p} - \widehat{\mathbf{x}}\|_2^2. \tag{15}$$

Here, $|\mathcal{P}_i|$ is the number of observed points, $\mathbf{p} \in \mathbb{R}^3$ is an observed point, and $\widehat{\mathbf{x}} \in \mathbb{R}^3$ is a predicted point. Fidelity measures the squared distance from each observation to its nearest predicted point.

The Minimum Matching Distance (MMD) compares the prediction with its closest reference shape:

$$\text{MMD}_i = \min_{\mathcal{X} \in \mathcal{R}} \left[\frac{1}{|\widehat{\mathcal{X}}_i|} \sum_{\widehat{\mathbf{x}} \in \widehat{\mathcal{X}}_i} \min_{\mathbf{x} \in \mathcal{X}} \|\widehat{\mathbf{x}} - \mathbf{x}\|_2^2 + \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} \min_{\widehat{\mathbf{x}} \in \widehat{\mathcal{X}}_i} \|\mathbf{x} - \widehat{\mathbf{x}}\|_2^2 \right]. \tag{16}$$

Here, $\mathcal{X} \in \mathcal{R}$ is a candidate complete reference shape, $|\widehat{\mathcal{X}}_i|$ and $|\mathcal{X}|$ are the corresponding point counts, and $\mathbf{x} \in \mathbb{R}^3$ is a reference point. We average both metrics over the KITTI evaluation set and scale them by 10^3.

C Additional Experiments

C.1 Pose-Protocol Comparison on PCN

We evaluated the same Stage-II checkpoint on the 1,200-sample PCN test split under two pose protocols: the original, unrotated point clouds and independently sampled rotations about the three coordinate axes. The rotated evaluation used angles in $[-180°, 180°]$ with seed 2026. Both evaluations used single-precision DLM-CR with an 80-iteration budget and a target converged fraction of 0.999. The category-wise XYZ errors under these protocols are reported in Table 8.

The two averages differed by 0.0625, or 0.61% of the rotated score. Seven category-wise differences were below 1% in magnitude, whereas Lamp differed by 5.22%. The comparison is therefore consistent with similar aggregate accuracy under these protocols. It neither establishes exact rotation equivariance nor rules out category-specific effects.

C.2 HAAR-UNIFORM $\mathrm{SO}(3)$ ROTATIONS ON PCN

Independent sampling of three axis angles is not uniform on $\mathrm{SO}(3)$. The additional protocol mapped three pseudorandom $\mathcal{U}(0,1)$ variables to a unit quaternion for each sample. The resulting Haar-uniform rotation was applied jointly to the partial input, ground truth, and anchors. Both protocols used seed 2026 and the same 1,200 PCN test samples; we did not rerun the standard-pose evaluation.

Table 9: Category-wise PCN results under independent three-axis and Haar-uniform $\mathrm{SO}(3)$ rotations. Each category contains 150 test samples. Values are mean XYZ CD-$L_1 \times 10^3$; lower is better. ESCAPE uses the authors' released checkpoint and SciPy LM solver. PA-RangeLM uses the Stage-II checkpoint and DLM-CR. Baseline [6] is ESCAPE (Bekci et al., 2025).

	Independent rotations		Haar-uniform rotations	
Category	ESCAPE (CVPR'25)	PA-RangeLM	ESCAPE (CVPR'25)	PA-RangeLM
Airplane	8.9752	**7.5871**	8.9044	**7.5784**
Cabinet	13.8866	**13.0399**	13.9902	**13.0405**
Car	10.7709	**10.0670**	10.7334	**10.0561**
Chair	11.1468	**10.3649**	11.1078	**10.3712**
Lamp	8.6133	**7.9296**	8.5513	**7.9174**
Sofa	14.4457	**13.9008**	14.4231	**13.9083**
Table	9.8660	**9.0871**	9.7493	**9.1033**
Watercraft	10.3603	**9.6071**	10.4006	**9.7010**
Average	11.0081	**10.1979**	10.9825	**10.2095**

Under independent three-axis rotations, the mean XYZ CD-$L_1 \times 10^3$ was 10.1979 for PA-RangeLM and 11.0081 for ESCAPE. Under Haar-uniform rotations, the corresponding values were 10.2095 and 10.9825. PA-RangeLM had a lower category mean in all eight categories under the Haar-uniform protocol. Changing the rotation sampler shifted the overall mean by $+0.0116$ for PA-RangeLM and -0.0256 for ESCAPE. These results apply to the evaluated checkpoints, test split, and seed, and do not establish exact rotation equivariance.

C.3 COORDINATE-RECOVERY SOLVER COMPARISON

This comparison reuses the checkpoint and rotation manifest described above. The raw Stage-I ranges, anchors, Cartesian targets, and solver weights are fixed between runs. The prototype branch, bounded range correction, and reliability head remain inactive, and both solvers start from the origin. ESCAPE uses its released point-wise SciPy LM implementation. DLM-CR uses the batched PyTorch implementation with 80 single-precision iterations. The benchmark script names this implementation `dcm_cr`; DLM-CR is the corresponding manuscript term.

For each recovered point, the final range residual is its point-to-anchor distance minus the predicted range. We report aggregate residual MAE and RMSE, together with mean XYZ CD-$L_1 \times 10^3$. The point-level failure rate follows solver-specific status tests. ESCAPE counts an unsuccessful SciPy termination or a non-finite solution. DLM-CR counts a point that does not meet its convergence criterion within 80 iterations. Sample-level numerical failure counts samples containing non-finite recovered coordinates. The point-level rates therefore do not represent an identical event.

Table 10: Coordinate recovery from the same frozen ESCAPE checkpoint on the independently rotated PCN test split. All values are taken from the completed evaluation logs. Point solver failure is the reported failure or non-convergence rate over 19,660,800 recovered points; sample numerical failure is the rate of non-finite outputs over 1,200 samples. Lower is better for every metric. Baseline [6] is ESCAPE (Bekci et al., 2025).

Solver	Residual MAE	Residual RMSE	XYZ CD-L_1 ($\times 10^3$)	Point failure (%)	Sample failure (%)
ESCAPE LM (CVPR'25)	$1.226502386 \times 10^{-3}$	$4.285543148 \times 10^{-3}$	11.008113	0.000209	0.000000
DLM-CR	$1.241169306 \times 10^{-3}$	$4.647583605 \times 10^{-3}$	11.025873	2.570989	0.000000

ESCAPE LM produced the lower residual MAE, residual RMSE, and mean XYZ error. The mean XYZ CD-$L_1 \times 10^3$ differed by 0.017760. DLM-CR returned finite coordinates for every sample, although 2.570989% of points did not meet its convergence criterion within 80 iterations. The different status tests prevent a direct reliability ranking from the point-level rates. This frozen-checkpoint comparison shows no residual or mean-XYZ advantage for DLM-CR on uncorrected

Stage-I ranges. It does not test bounded correction, uncertainty weighting, or end-to-end Stage-II training.

C.4 Point-Level Error Ranking by Relative Precision

We tested whether the relative precision weights from the Stage-II uncertainty head rank errors in the recovered points. The evaluation used the Stage-II checkpoint and the full 1,200-sample PCN test split. We applied independent three-axis rotations with seed 2026 and recovered coordinates with 80 single-precision DLM-CR iterations and an initial damping of 10^{-3}. All recovered points had finite coordinates.

The evaluation covered 19,660,800 predicted points. Of these, 2.7110% did not meet the DLM-CR stopping criterion within the 80-iteration budget.

The K relative weights for a point have a fixed sum. Their arithmetic mean is therefore constant and cannot rank points. We derive a point score from the effective anchor count:

$$n_i^{\mathrm{eff}} = \frac{\left(\sum_{k=1}^{K} w_{ik}\right)^2}{\sum_{k=1}^{K} w_{ik}^2}, \qquad u_i = 1 - \frac{n_i^{\mathrm{eff}}}{K}. \tag{17}$$

Concentrating the precision on fewer anchors decreases n_i^{eff} and increases u_i.

For recovered point $\widehat{\mathbf{x}}_i$, we define the target error as the one-sided nearest-neighbor distance $e_i = \min_{\mathbf{x}\in\mathcal{X}_{\mathrm{gt}}} \|\widehat{\mathbf{x}}_i - \mathbf{x}\|_2$. Spearman correlation measures the rank association between u_i and e_i. AUROC measures detection of the largest-error 10% of points within each shape. For risk–coverage analysis, we retain points from low to high u_i and average their errors.

We computed each metric within a shape and then macro-averaged over the 1,200 shapes. The reported 95% confidence intervals are percentile bootstrap intervals over shapes from 2,000 resamples.

The point score was positively associated with error (Spearman's $\rho = 0.2859$, 95% CI $[0.2742, 0.2977]$). Its AUROC for the highest-error 10% of points was 0.7503 (95% CI $[0.7430, 0.7567]$). Sorting by u_i gave an area under the risk–coverage curve (AURC) of 6.4196×10^{-3} (95% CI $[6.1744, 6.7308] \times 10^{-3}$). The expected risk under random retention was 8.3290×10^{-3}. Relative to that reference, the AURC was 22.91% lower (95% CI $[21.99\%, 23.80\%]$). Table 11 gives selected coverage levels.

Table 11: Point-level risk under selective retention on the independently rotated PCN test set. Coverage is the fraction of points retained after sorting from low to high u_i. Risk is the macro-averaged one-sided nearest-target error scaled by 10^3, not bidirectional CD-L_1. Random risk is the expected error under random retention. Lower is better.

Coverage (%)	Uncertainty-ranked risk ($\times 10^3$)	Random-ranking risk ($\times 10^3$)	Risk reduction (%)
10	5.7368	8.3290	31.12
50	6.1316	8.3290	26.38
80	7.1125	8.3290	14.61
90	7.6403	8.3290	8.27
100	8.3290	8.3290	0.00

Spearman correlation was positive for 93.3% of shapes, AUROC exceeded 0.5 for 96.3%, and AURC was lower than random retention for 96.1%. The mean correlation was positive and the mean AUROC exceeded 0.5 in every category. Airplane had the lowest category values ($\rho = 0.1501$, AUROC 0.6557, 13.46% AURC reduction). Sofa had the highest ($\rho = 0.3769$, AUROC 0.7953, 31.17% AURC reduction).

We also compared each relative range weight with the absolute error of its point–anchor range. This measurement-level association was weak. Spearman's ρ was 0.0665 and AUROC was 0.5689. Its AURC was 0.38% higher than random retention (95% CI $[0.02\%, 0.74\%]$ higher). These results do not support interpreting an individual weight as a calibrated range-error estimate. By contrast, the aggregated point score had lower mean AURC than random retention.

The risk in Table 11 is one-sided. Removing predicted points also changes the target-to-prediction term, so these values are not comparable to the bidirectional CD-L_1 values in the main tables. The

weights affect both DLM-CR and the score u_i. The analysis therefore tests whether u_i usefully ranks outputs from this pipeline, not whether it provides an independent probabilistic calibration.

C.5 MVP Categories Absent from PCN Training

The PCN-trained ESCAPE and PA-RangeLM checkpoints were evaluated on the remaining MVP categories: bed, bench, bookshelf, bus, guitar, motorbike, pistol, and skateboard. None occurs in the PCN training set, and neither model was trained or fine-tuned on MVP. Both runs used the same 10,400 partial–complete pairs. Following the main PCN protocol, the three rotation angles were sampled independently from $[-180°, 180°]$ with seed 2026.

Table 12: PCN-to-MVP completion on categories absent from PCN training. Each category contains 1,300 partial–complete pairs from all 26 views. Values are mean XYZ CD-$L_1 \times 10^3$; lower is better. Both columns use the same rotation manifest, PyTorch LM implementation, and 80-iteration budget. Baseline [6] is ESCAPE (Bekci et al., 2025).

Category	ESCAPE (CVPR'25)	PA-RangeLM
Bed	16.3177	**16.2012**
Bench	11.0070	**10.1100**
Bookshelf	15.3180	**14.0553**
Bus	14.8782	**13.5789**
Guitar	8.2120	**7.3167**
Motorbike	11.5382	**10.8608**
Pistol	11.6644	**10.6688**
Skateboard	11.3119	**10.3897**
Average	12.5309	**11.6477**

PA-RangeLM had a lower mean than ESCAPE in each evaluated category. The eight-category average was 11.6477 for PA-RangeLM and 12.5309 for ESCAPE, a difference of 0.8832 (7.0%). These values come from one checkpoint per model and one rotation seed under the specified evaluation protocol.

The data source and category set both differ from PCN. This experiment therefore does not isolate category transfer from dataset shift, and it is not an open-set recognition evaluation.

C.6 Test-Time LM Iteration Sensitivity

We varied the DLM-CR iteration budget while holding the Stage-II checkpoint and test protocol fixed. Each run used the 1,200 independently rotated PCN test samples with seed 2026, batch size four, single-precision recovery, and an initial damping of 10^{-3}. Table 13 reports the resulting accuracy, solver cost, and failure statistics.

Table 13: Sensitivity to the test-time DLM-CR iteration budget on independently rotated PCN. XYZ CD-L_1 is scaled by 10^3; lower is better. Solver time is the synchronized mean per sample. Point non-convergence denotes points that did not satisfy the stopping criterion within the stated budget. Sample failure denotes samples containing a non-finite recovered coordinate. GPU memory is peak allocated memory during evaluation.

LM iterations	XYZ CD-L_1 ($\times 10^3$)	Solver time (ms/sample)	Point non-convergence (%)	Sample failure (%)	Peak GPU memory (MB)
10	10.4163	29.38	12.650350	0.000000	306.26
20	10.2197	56.44	3.165426	0.000000	306.26
40	10.1996	111.88	2.740046	0.000000	306.26
80	10.1979	240.63	2.711019	0.000000	306.26

Increasing the budget from 10 to 20 iterations reduced XYZ CD-L_1 by 0.1966 and point non-convergence by 9.4849 percentage points. The CD changes from 20 to 40 and from 40 to 80 iterations were 0.0201 and 0.0017, respectively. Solver time increased from 29.38 to 240.63 ms per sample across the tested range. All runs returned finite coordinates for every sample, and peak allocated memory remained 306.26 MB. We retained 80 iterations for the other evaluations. These measurements apply to this checkpoint, seed, batch size, and device.

C.7 Training-Time Differentiable LM Iteration Sensitivity

We next varied the number of differentiable LM iterations used in Stage-II training. All runs used five no-gradient warm-up iterations, 4,096 sampled training points, the same Stage-I initialization, seed 2026, and 50 epochs. Each selected checkpoint was evaluated with 80 DLM-CR iterations on the same independently rotated PCN test set.

Table 14: Sensitivity to the number of differentiable LM iterations in Stage-II training. XYZ CD-L_1 is measured on the 1,200-sample independently rotated PCN test set and scaled by 10^3; lower is better. Training time is the W&B process runtime through final validation for 50 epochs. Peak GPU memory is the maximum device-level allocated memory sampled by the W&B system monitor. Each row is one training run.

Differentiable LM iterations	Test XYZ CD-L_1 ($\times 10^3$)	Training time (h)	Peak GPU memory (GiB)
1	10.2988	13.16	10.14
3	**10.1979**	15.03	9.94
5	10.2476	15.85	10.29

Three differentiable iterations gave the lowest test error among the evaluated settings. Relative to this configuration, one iteration reduced the recorded training time by 1.87 h but increased XYZ CD-L_1 by 0.1008. Five iterations increased training time by 0.82 h and produced a CD value 0.0496 higher. The sampled memory peaks ranged from 9.94 to 10.29 GiB. Because each setting was trained once, the timing and memory values describe these runs rather than run-to-run variability.

D Code and Data Availability

Code availability. During peer review, we will provide an anonymous code package containing the training and inference implementation, configuration files, and evaluation scripts needed to reproduce the reported experiments. The package will also document checkpoint selection and the random-rotation manifests. A permanent public repository identifier will be added to the camera-ready version.

Data availability. This study reuses the public PCN (Yuan et al., 2018), MVP (Pan et al., 2021), and KITTI (Geiger et al., 2012) datasets under their original access conditions; no new primary dataset is introduced. The official MVP completion benchmark and its access conditions are identified in Appendix A.3. The anonymous code package will identify the exact splits and preprocessing used here and will include the per-sample evaluation outputs and rotation metadata underlying the reported aggregate results.

E QUALITATIVE VISUALIZATIONS

E.1 PCN RESULTS

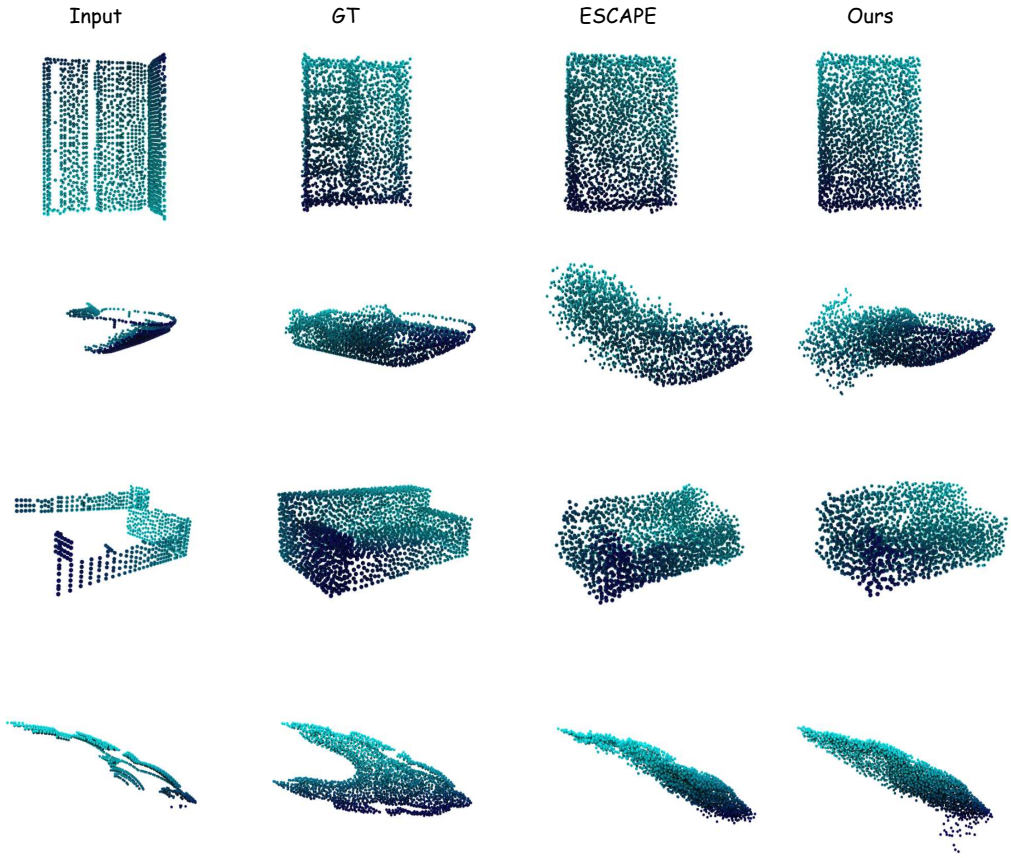

Figure 4: Additional qualitative results on PCN.

E.2 MVP Categories Shared with PCN

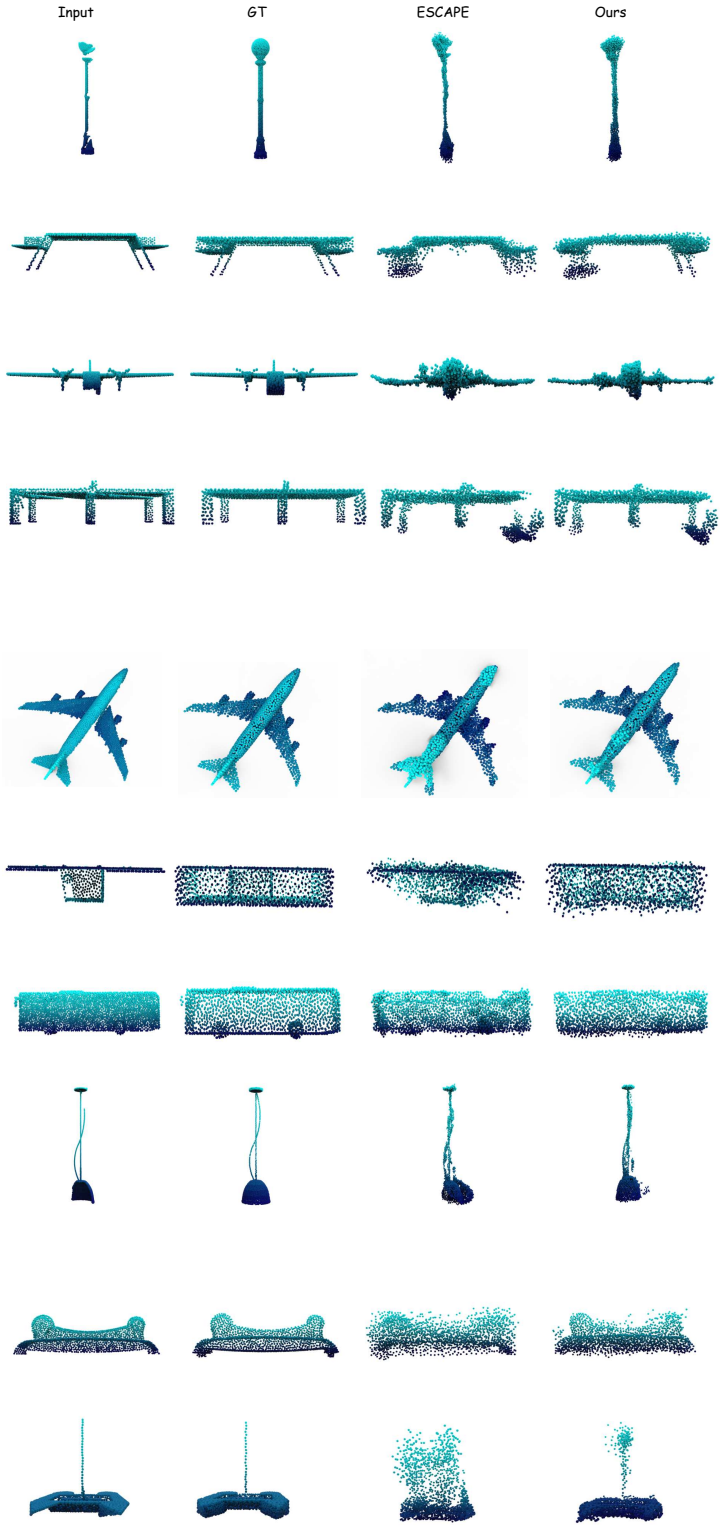

Figure 5: Additional MVP results for categories shared with PCN.

E.3 MVP CATEGORIES ABSENT FROM PCN

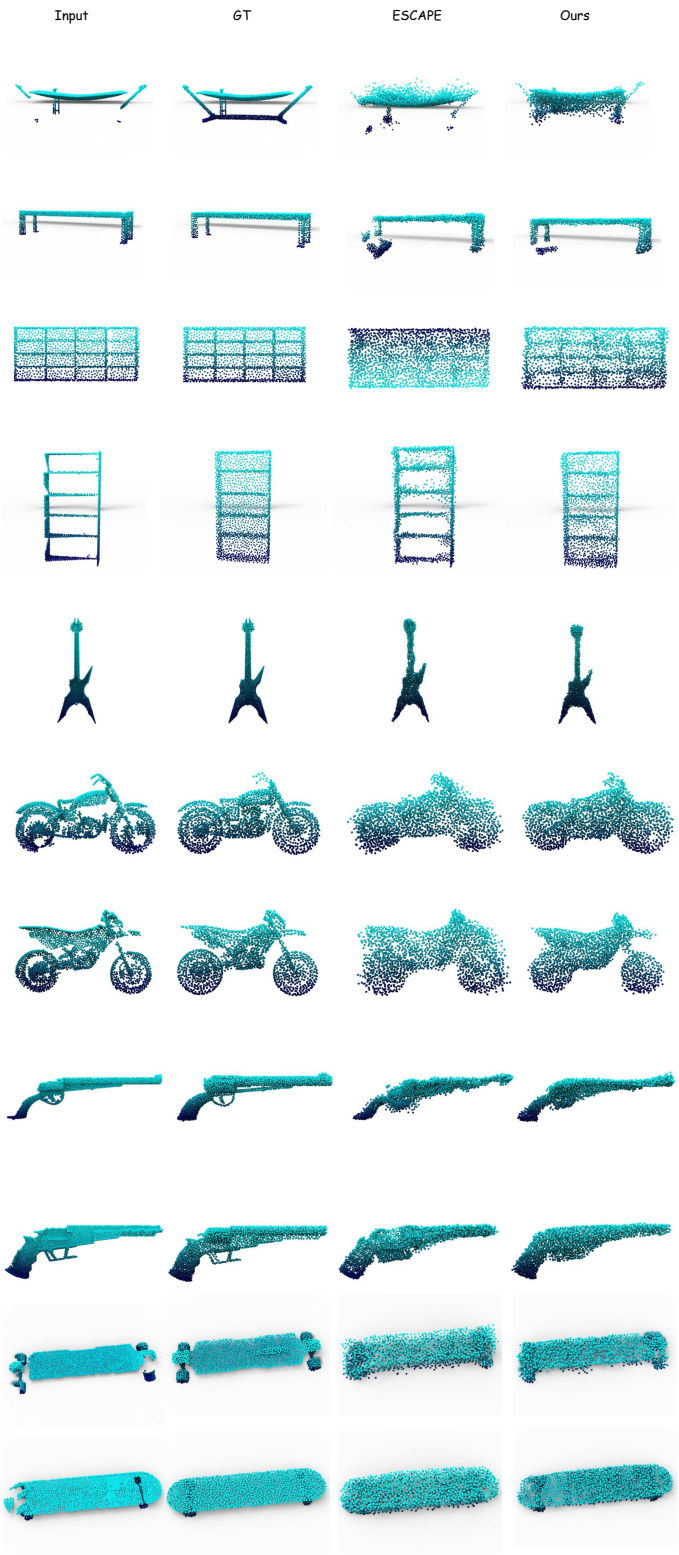

Figure 6: MVP results for the eight categories absent from PCN training.

E.4 KITTI RESULTS



Figure 7: Additional qualitative results on KITTI.