PRISM: Industrial Point Cloud Completion via Meta-Learned Test-Time Self-Supervised Adaptation

Seung Yeop Ha[1,2], Jae Hun Hwang[1,3], Jong Pil Yun[1,4], Seung-Kyum Choi[5], Jun-Geol Baek[2], Hong-In Won[1,*]
[1]Manufacturing AI Research Center, Korea Institute of Industrial Technology, Incheon, Korea
[2]Dept. of Industrial and Management Engineering, Korea University, Seoul, Korea
[3]Dept. of Artificial Intelligence Convergence, Hanyang University, Seoul, Korea
[4]Graduate School of Advanced Imaging Science, Multimedia & Film, Chung-Ang University, Seoul, Korea
[5]G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA
[*]Corresponding Author: luvhayym@kitech.re.kr
Email: {chris257, jv29734v, rebirth, luvhayym}@kitech.re.kr, schoi@me.gatech.edu, jungeol@korea.ac.kr

***Abstract*—Low-cost LiDAR is increasingly adopted for 3D perception in smart manufacturing, but sparse returns, occlusions, and varying sensor configurations can degrade point cloud completion and downstream safety envelope estimation. Most completion methods rely on sparse point-set losses such as Chamfer Distance, offering weak surface supervision under severe sparsity and occlusion. We propose PRISM (Projection-Regularized Industrial Self-supervised Meta-adaptation), a LiDAR point cloud completion framework that couples completion with projection-based depth constraints and episodic meta-learning over auxiliary geometric objectives. In meta-training, PRISM learns an initialization for fast test-time adaptation and meta-optimizes adaptive auxiliary loss weights. At test time, self-supervised adaptation enforces projection-based depth consistency with the same geometric regularizers. Experiments on ShapeNet and our industrial RoboArm-Depth dataset demonstrate improved reconstruction and robot geometry recovery under severe sensor-induced sparsity and occlusion, supporting more reliable LiDAR-based safety envelope estimation.**

***Index Terms*—LiDAR occlusion, point cloud completion, meta-learning, test-time adaptation, manufacturing AI**

I. Introduction

Light Detection and Ranging (LiDAR)-based 3D perception systems are increasingly fundamental to automation, safety monitoring, and quality assurance in modern manufacturing [1], [2]. Industrial applications span autonomous mobile robot navigation, process monitoring via overhead sensor arrays, and digital twin construction [2], [3]. The adoption of low-cost LiDAR units enables scalable 3D sensing across factory environments; however, dense machinery, specular materials, and complex layouts often yield sparse point clouds with sensor-induced occlusions and blind spots that may conceal collision risks [2], [4].

A common response to sparse LiDAR is point cloud completion, which predicts missing geometry from partial observations [5], [6]. Yet, under severe sparsity and occlusion, many completion pipelines are supervised mainly by sparse point-set distances such as Chamfer Distance (CD), which are insensitive to density distribution [7], [8] and provide weak surface-level guidance. To strengthen geometric supervision, we couple point cloud completion with projection-based depth supervision from a single calibrated LiDAR viewpoint, providing dense per-pixel cues that complement sparse point-set losses, particularly near occlusion boundaries.

This work was supported by the Korea Institute of Industrial Technology (KITECH) under Grant No. EH-26-0002, and by the Korea Evaluation Institute of Industrial Technology (KEIT) under Grant No. RS-2024-00423589.

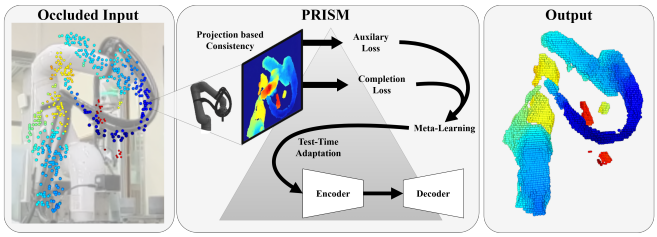

Fig. 1. PRISM recovers complete robot geometry from sparse occluded industrial LiDAR via meta-learned test-time adaptation with projection-depth consistency.

Beyond sparsity, industrial LiDAR deployments face domain shifts caused by sensor relocation and equipment reconfiguration. Recent advances in test-time adaptation (TTA) for 3D perception show promise [9]–[11], but robust adaptation under severe sparsity and unpredictable variation remains challenging. This motivates frameworks that leverage projection-based depth constraints to provide dense geometric signals and adapt efficiently at test time without labels.

To this end, we propose **PRISM** (**P**rojection-**R**egularized **I**ndustrial **S**elf-supervised **M**eta-adaptation) (Fig. 1), an industrial LiDAR point cloud completion framework that integrates projection-based depth supervision with episodic meta-learning for fast test-time adaptation.

Our main contributions are:

- We formalize industrial LiDAR completion with projection-based depth consistency to address weak surface guidance from sparse point-set supervision under severe sparsity and occlusion.
- We propose PRISM, which couples projection-based depth supervision with episodic meta-learning to learn adaptive auxiliary loss weights, enabling fast test-time adaptation without labels.

- We demonstrate state-of-the-art completion accuracy in Chamfer Distance on ShapeNet and strong performance on our industrial RoboArm-Depth dataset, with notable gains in robot geometry recovery under sensor-induced occlusions.

II. Related Work

A. Industrial LiDAR Occlusion and Domain Shift

Industrial LiDAR operates in cluttered factory cells with machinery, reflective and glossy materials, and frequently reconfigured sensor placements, which together induce severe sparsity and complex self-occlusion patterns around robots and equipment [2], [4]. Unlike automotive or indoor benchmarks, factory deployments must cope with low-cost sensors, material-dependent signal loss, and rapidly changing layouts, yielding continual domain shifts over time [12]. Collaborative and human–robot workcells are particularly challenging: robot arms self-occlude their own links and are intermittently hidden by fixtures or workpieces, creating large missing regions in the sensed scene and increasing collision risk [13], [14].

Recent surveys on industrial point cloud perception and digital-twin integration highlight substantial gaps between standard driving/RGB-D datasets and real manufacturing environments, including differences in sensor geometry, reflectance statistics, clutter levels, and occlusion [1], [2]. These characteristics motivate completion frameworks designed for industrial LiDAR. Such frameworks must recover robot and equipment geometry under extreme sparsity and structured self-occlusion, while remaining robust to reconfiguration-driven shifts [2], [4].

B. Learning-based Point Cloud Completion

Point cloud completion reconstructs complete 3D structures from partial observations using encoder–decoder architectures, point-based backbones such as PointNet++ [15], graph- and transformer-based networks, and generative priors, with recent surveys systematizing architectures and benchmarks across synthetic objects and scene-level datasets [2], [16]. Representative networks such as PCN [5], PoinTr [6], SnowflakeNet [17], HDANet [18], PST-Net [19], and HSPC-Net [20] achieve strong performance on ShapeNet-style benchmarks, while more recent approaches including FSC [21], PCDreamer [22], and IE-PMMA [23] explore few-point completion and diffusion/generative models.

However, most approaches rely on sparse point-set supervision such as CD [5], [6], [17] and optimize completion largely in 3D point space, which is known to be insensitive to density distribution and less effective under extreme sparsity, strong occlusion, and highly non-uniform point density typical of factory LiDAR [7]. In addition, depth estimation and 3D completion are often treated as separate problems [16], making it non-trivial to inject dense, per-pixel geometric cues that are especially informative near occlusion boundaries. These limitations motivate unified formulations that couple completion with projection-based depth constraints. By projecting 3D points onto a 2D image plane, depth maps aggregate sparse observations into a dense grid structure that provides spatially coherent geometric signals, around occlusion boundaries where point density is most irregular, thereby strengthening surface supervision under industrial sparsity patterns.

C. Meta-learning and Test-time Adaptation for 3D Perception

Meta-learning aims to train models that can rapidly adapt to new tasks or domains with limited supervision, for example by learning an initialization that is effective after a few gradient steps as in model-agnostic meta-learning [24]. In 3D vision, meta-learning has been explored for cross-domain point cloud classification, registration, and completion, including recent work on domain-adaptive point cloud completion [9]. TTA, in contrast, updates model parameters at inference using unlabeled test observations to address distribution shifts; entropy-minimization methods such as TENT [25] adjust selected parameters based on prediction statistics, and continual adaptation has been investigated for LiDAR segmentation and multi-task point cloud understanding under domain shift [10], [12].

More recently, several frameworks have combined meta-learning with test-time training, showing that meta-learned initializations or auxiliary objectives can improve the effectiveness and stability of self-supervised adaptation to unseen distributions [11], [26]. Building on this line, PRISM integrates projection-based depth cues into completion and uses episodic meta-training to (i) meta-learn an initialization for fast TTA and (ii) meta-optimize loss-weight logits that adaptively balance auxiliary geometric objectives. At test time, PRISM performs self-supervised adaptation with projection/depth consistency and the same geometric regularizers, improving robustness under industrial LiDAR sparsity and domain shifts.

III. Method

A. Overview

PRISM adopts a two-stream design for industrial LiDAR completion (Fig. 2). The point-cloud stream encodes the partial input $\mathcal{P}$ using a CRA-PCN encoder and predicts the completed point set $\hat{\mathcal{C}}$ via coarse-to-fine decoding. In parallel, the depth stream encodes sparse depth from projecting $\mathcal{P}$ under a calibrated view using a ResNet encoder, providing dense geometric signals that complement point supervision. The two streams remain decoupled at the feature level and interact only through geometry-aware auxiliary objectives, enabling modular training and TTA. A refinement module further sharpens local geometry within the normalized workspace.

B. Problem Setup and Projection Operator

We study sparse LiDAR point cloud completion in industrial scenes. Given a partial point cloud $\mathcal{P} = \{p_i\}_{i=1}^{N}$ with $p_i \in [-1, 1]^3$ in a normalized workspace, the goal is to predict a dense completed cloud $\hat{\mathcal{C}} = \{\hat{c}_j\}_{j=1}^{N_{\text{out}}}$. When available during training, the ground-truth complete cloud is denoted by $\mathcal{C}$.

To obtain dense depth supervision, we assume a single calibrated view per frame, consistent with deployment constraints under extreme sparsity. Let $K \in \mathbb{R}^{3\times3}$ denote the camera intrinsics and (R_v, t_v) the extrinsics of view v. A point p_i

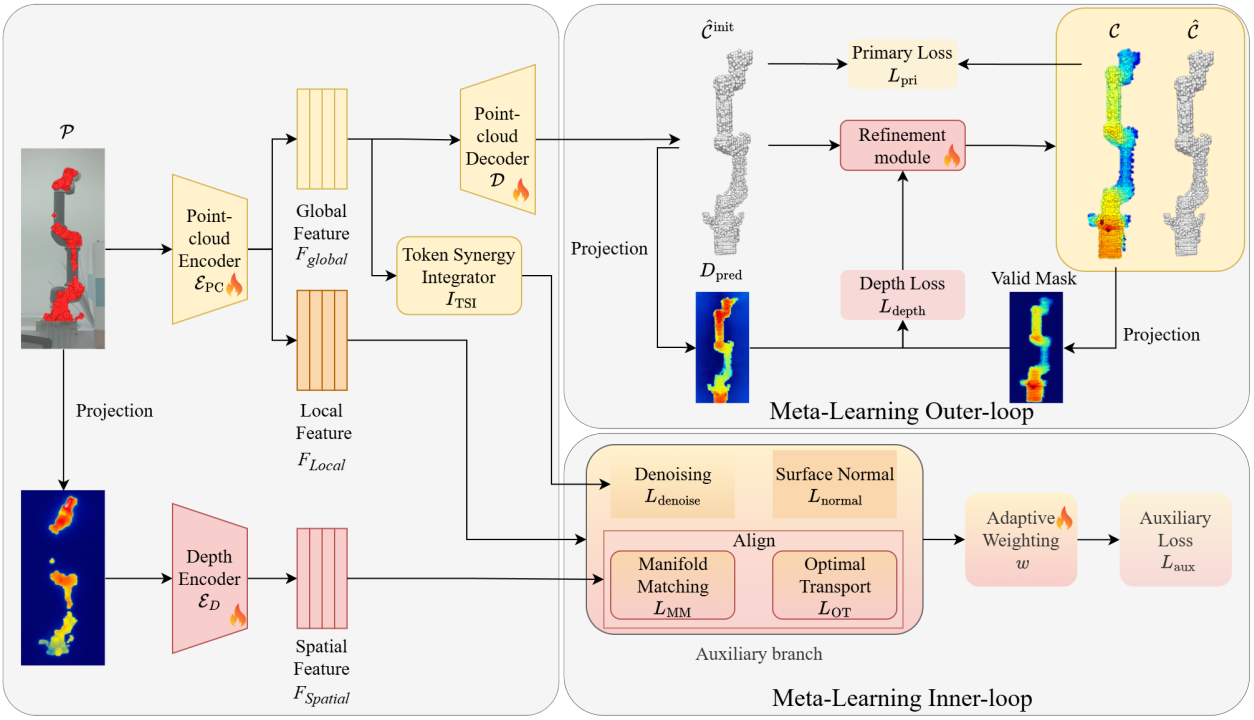

Fig. 2. Overview of PRISM. A point-cloud stream completes the partial input $\mathcal{P}$ into $\hat{\mathcal{C}}$, a depth stream encodes the projected sparse depth, and auxiliary objectives couple the streams while a bounded refinement module adjusts local geometry within the normalized workspace.

is transformed into the camera frame as $x_i = R_v p_i + t_v$, and its depth is $z_i = (x_i)_z$. We apply perspective projection $\pi([x, y, z]^\top) = (x/z,\, y/z)$, multiply by the intrinsic matrix K, and discretize to integer pixel coordinates. The depth map $D_v \in \mathbb{R}^{H\times W}$ and valid-pixel mask $M_v \in \{0,1\}^{H\times W}$ are rendered via z-buffering. Let $\mathcal{I}_v(u)$ denote the set of point indices whose projection falls on pixel u; the rendering operator is formalized in Eq. 1.

$$D_v(u) = \min_{i\in\mathcal{I}_v(u)} z_i,$$
$$M_v(u) = \begin{cases} 1 & \text{if } \mathcal{I}_v(u) \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

C. Completion Backbone and Bounded Refinement

The point-cloud stream uses an encoder–decoder backbone producing coarse-to-fine predictions with equal supervision across stages to preserve a strong completion prior under severe occlusion. To improve surface fidelity, we append a lightweight refinement module. For each point $\hat{c}_j$, it takes point coordinates and decoder features ϕ_j as input and outputs a bounded offset via an MLP g. The offset is constrained with tanh activation and clamping, as formalized in Eq. 2.

$$\Delta_j = \delta_{\max}\,\tanh\big(g(\hat{c}_j, \phi_j)\big),$$
$$\hat{c}_j^{\text{ref}} = \text{clamp}(\hat{c}_j + \Delta_j,\, -1,\, 1)\,, \tag{2}$$
$$\hat{\mathcal{C}} = \{\hat{c}_j^{\text{ref}}\}_{j=1}^{N_{\text{out}}}.$$

a) Projection-based Depth and Consistency.: Let $\Pi(\cdot)$ denote the rendering operator in Eq. 1 under the calibrated view, returning a depth map and valid-pixel mask. During training, we supervise the refinement module by matching the rendered depth of the refined prediction to the ground-truth complete cloud over commonly valid pixels. We obtain $(D_{\text{pred}}, M_{\text{pred}}) = \Pi(\hat{\mathcal{C}})$ and $(D_{\text{gt}}, M_{\text{gt}}) = \Pi(\mathcal{C})$, and define $\Omega_{\text{sup}} = \{u \mid M_{\text{pred}}(u) = 1,\ M_{\text{gt}}(u) = 1\}$ as shared valid pixels. To avoid destabilizing the completion backbone, this loss updates only refinement parameters, as in Eq. 3.

$$L_{\text{depth}}^{\text{sup}} = \frac{1}{|\Omega_{\text{sup}}| + \epsilon} \sum_{u\in\Omega_{\text{sup}}} \big|D_{\text{pred}}(u) - D_{\text{gt}}(u)\big|. \tag{3}$$

At test time, $\mathcal{C}$ is unavailable, so we enforce depth consistency with the observed projected depth from the input. We compute $(D_{\text{in}}, M_{\text{in}}) = \Pi(\mathcal{P})$ and define $\Omega_{\text{cons}} = \{u \mid M_{\text{in}}(u) = 1,\ M_{\text{pred}}(u) = 1\}$. This consistency term constrains the predicted geometry to agree with visible structure while leaving unobserved regions governed by the learned completion prior, as formalized in Eq. 4.

$$L_{\text{depth}}^{\text{cons}} = \frac{1}{|\Omega_{\text{cons}}| + \epsilon} \sum_{u\in\Omega_{\text{cons}}} \big|D_{\text{pred}}(u) - D_{\text{in}}(u)\big|. \tag{4}$$

D. Geometry-aware Self-supervised Objectives

a) Surface-normal regularization.: We predict per-point surface normals $\hat{n}_j$ on $\hat{\mathcal{C}}$. As pseudo-targets, we compute principal component analysis (PCA) normals n_j^{pca} from local neighborhoods of $\hat{\mathcal{C}}$ and stop gradients through these anchors. Since PCA normals have sign ambiguity, we use a sign-

invariant agreement term and encourage local smoothness on a neighbor graph $\mathcal{E}$ over $\hat{\mathcal{C}}$, as in Eq. 5.

$$L_{\text{normal}} = \frac{1}{N_{\text{out}}} \sum_{j} \Big(1 - \big|\hat{n}_j^{\top} n_j^{\text{pca}}\big|\Big) + \lambda_{\text{s}} \frac{1}{|\mathcal{E}|} \sum_{(j,k)\in\mathcal{E}} \|\hat{n}_j - \hat{n}_k\|_1 . \tag{5}$$

b) Denoising.: To improve robustness to structured sparsity and sensor artifacts, we train a denoising branch that maps a perturbed input to a geometrically consistent point set. Let $\tilde{\mathcal{P}} = \mathcal{A}(\mathcal{P})$ be an augmented version of $\mathcal{P}$ obtained by small perturbations, and let the denoiser predict $\mathcal{P}^{\text{dn}} = \tilde{\mathcal{P}} + \Delta^{\text{dn}}$. We supervise denoising using a self-reconstruction objective to an upsampled target $\mathcal{U}(\mathcal{P})$ together with a small offset penalty. This loss encourages the denoising branch to undo small corruptions while preserving the underlying observed structure in $\mathcal{P}$, as formalized in Eq. 6.

$$L_{\text{denoise}} = \text{CD}\big(\mathcal{P}^{\text{dn}}, \mathcal{U}(\mathcal{P})\big) + \lambda_{\Delta} \frac{1}{|\tilde{\mathcal{P}}|} \sum_{i} \big\|\Delta_i^{\text{dn}}\big\|_2^2 . \tag{6}$$

c) 3D–depth alignment.: The depth stream encodes the projected sparse depth, and the point-cloud stream produces 3D features. We align these representations without explicit fusion by combining a global manifold-matching term L_{MM} and a local optimal transport term L_{OT}.

For global alignment, let F^{3D} and F^{D} denote pooled features from the point-cloud and depth streams. The manifold-matching loss compares their first- and second-order statistics, where $\mu(\cdot)$ and $\Sigma(\cdot)$ are the empirical mean and covariance over the batch. This loss acts as a tractable proxy for reducing the discrepancy between the two latent feature manifolds under severe sparsity. For local alignment, we extract tokens from the 3D and depth feature maps, $\mathcal{T}^{\text{3D}} = \{a_i\}_{i=1}^{m}$ and $\mathcal{T}^{\text{D}} = \{b_j\}_{j=1}^{n}$, construct the cost matrix $C_{ij} = \|a_i - b_j\|_2^2$, and obtain an entropically regularized transport plan T via the Sinkhorn operator. The overall alignment loss combines both terms to encourage consistent cross-modal correspondence at global and token levels while remaining differentiable and stable under sparse observations, as formalized in Eq. 7–9.

$$L_{\text{MM}} = \big\|\mu(F^{\text{3D}}) - \mu(F^{\text{D}})\big\|_2^2 + \lambda_{\Sigma} \big\|\Sigma(F^{\text{3D}}) - \Sigma(F^{\text{D}})\big\|_F^2 , \tag{7}$$

$$T = \text{Sinkhorn}(C; \epsilon_{\text{OT}}), \qquad L_{\text{OT}} = \langle T, C\rangle \;=\; \sum_{i=1}^{m}\sum_{j=1}^{n} T_{ij}\, C_{ij}, \tag{8}$$

$$L_{\text{align}} = L_{\text{MM}} + \lambda_{\text{OT}}\, L_{\text{OT}}. \tag{9}$$

E. Meta-learned Weighting and Training Objective

a) Primary loss.: We train the completion backbone using a coarse-to-fine Chamfer objective. Let $\hat{\mathcal{C}}^{(s)}$ denote the stage-s prediction with N_s points. To ensure fair comparison across stages with varying point counts, the ground-truth complete cloud $\mathcal{C}$ is downsampled to match each stage's resolution using farthest point sampling. The primary loss applies equal supervision to all stages, enabling strong gradients for coarse predictions under severe occlusion, as formalized in Eq. 10.

$$L_{\text{pri}} = \sum_{s=1}^{S} \text{CD}\Big(\hat{\mathcal{C}}^{(s)},\, \text{FPS}(\mathcal{C}, N_s)\Big) . \tag{10}$$

b) Meta-learning auxiliary weights.: We meta-learn the relative importance of auxiliary objectives using learnable logits $\eta \in \mathbb{R}^3$ and normalized weights $w = \text{softmax}(\eta)$. To ensure comparable scales, each auxiliary loss is normalized before weighting. During training, the auxiliary modules remain frozen and compute only self-supervised losses. The encoder receives gradients from both the primary completion loss and the weighted auxiliary objective through a two-stage process. In the inner loop, the encoder is adapted using the weighted auxiliary loss. In the outer loop, both the encoder and decoder are updated using the primary completion loss, and the encoder additionally receives gradients from the weighted auxiliary loss. To preserve zero-shot capability, we include a zero-step term using the un-adapted encoder. The weights w are updated only by the auxiliary loss using a separate learning rate, as formalized in Eq. 11-12.

$$L_{\text{aux}} = w_1 L_{\text{normal}} + w_2 L_{\text{denoise}} + w_3 L_{\text{align}}, \tag{11}$$

$$\min_{\theta_{\text{enc}},\theta_{\text{dec}},\eta} (1-\beta_0)\,[L_{\text{pri}}(\theta_{\text{enc}},\theta_{\text{dec}}) + \lambda_{\text{aux}}\, L_{\text{aux}}(\theta_{\text{enc}},\eta)] + \beta_0\, L_{\text{pri}}(\theta_{\text{enc}}). \tag{12}$$

F. Test-time Objective and Update

At test time, we minimize a label-free objective that combines projection-based depth consistency with the meta-learned auxiliary objectives (Fig. 3). Using the same rendering operator $\Pi(\cdot)$, we compute $L_{\text{depth}}^{\text{cons}}$ as in Eq. 4. We keep the completion decoder, refinement module, and auxiliary modules frozen during test-time adaptation. Only the encoder parameters are updated, which stabilizes adaptation and prevents drift in the completion prior while maintaining stable geometric priors from the frozen auxiliary modules.

Let η^* denote the meta-learned logits after training and $w^* = \text{softmax}(\eta^*)$. For each incoming frame, the adaptation procedure operates as follows: (i) we compute the test-time objective using the current encoder parameters and the frozen auxiliary modules, (ii) perform K_{TTA} gradient steps to update only the encoder parameters, (iii) forward the partial input through the adapted encoder and fixed decoder to produce the final completion $\hat{\mathcal{C}}$. The test-time objective reuses the meta-learned weights without modification, as formalized in Eq. 13.

$$L_{\text{TTA}} = L_{\text{depth}}^{\text{cons}} + w_1^* L_{\text{normal}} + w_2^* L_{\text{denoise}} + w_3^* L_{\text{align}}. \tag{13}$$

IV. Experiments

A. Datasets and Evaluation Protocol

a) ShapeNet.: We follow the ShapeNet-55 [27] benchmark protocol used in recent completion work. Each sample consists of a 2,048-point partial cloud and a 16,384-point complete cloud generated via viewpoint-based occlusion.

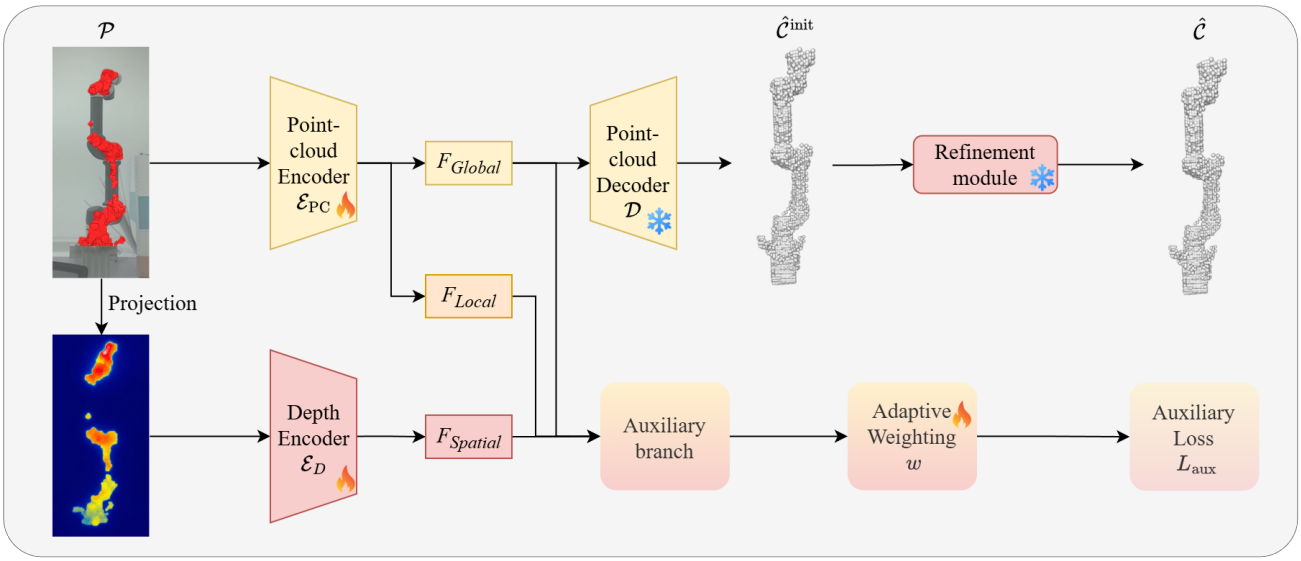

Fig. 3. Test-time self-supervised adaptation in PRISM. For each frame, only the point-cloud and depth encoders are updated using label-free objectives, while the completion decoder, refinement module, and auxiliary modules remain frozen with meta-learned weights.

b) RoboArm-Depth.: RoboArm-Depth is an industrial LiDAR dataset from collaborative robot cells with severe sensor-induced sparsity, LiDAR occlusions, and self-occlusions in cluttered workspaces. Each frame provides an L515 depth map and the corresponding partial point cloud, together with a completed point cloud acquired from an Ouster LiDAR, capturing occluded links, end-effectors, and fixtures. It comprises 20 captured sequences used exclusively for cross-domain evaluation, with no fine-tuning on target-domain labels.

We compare PRISM against representative point cloud completion baselines listed in Table I, using official implementations and hyperparameters when available or faithful reimplementations otherwise. All methods are trained on the same splits with identical input and output resolutions as PRISM. The primary evaluation metric is the CD between predicted and ground-truth point clouds, reported as CD $\times 10^{-4}$, with F-score at threshold 0.01 additionally reported to quantify surface coverage. Models are trained and evaluated on the official splits for ShapeNet. For RoboArm-Depth, we report cross-domain generalization by training PRISM on ShapeNet and testing on RoboArm-Depth without fine-tuning, after normalizing all point clouds to the $[-1, 1]^3$ cube, demonstrating the framework's ability to adapt to industrial scenes.

B. Implementation Details

All experiments use PyTorch on a single NVIDIA H200 GPU. Partial point clouds $\mathcal{P}$ have 2,048 points on ShapeNet following the standard occlusion protocol. The completion stream uses a CRA-PCN encoder–decoder producing predictions at 1,024, 2,048, 4,096, and 16,384 points, and we use the final output $\hat{\mathcal{C}}^{(4)}$ as the completion. Sparse depth maps are generated by projecting partial point clouds at 128×128 resolution and rendering per-pixel nearest depth with fixed intrinsics and extrinsics; invalid pixels are masked out. The depth stream uses a ResNet encoder without a decoder, interacting with the completion stream only through projection-based and auxiliary losses. Auxiliary branches for normal prediction, denoising, and cross-modal alignment share information via $\mathrm{I}_{\mathrm{TSI}}$, which pools features into an $m \times d$ token matrix, and a refinement module predicts bounded offsets with $\delta_{\max} = 0.1$.

Training uses SGD with learning rate $\gamma = 5 \times 10^{-5}$ and batch size 32 for 200 epochs on ShapeNet. The inner loop adapts the encoder using the weighted auxiliary loss from frozen modules, and the outer loop updates encoder and decoder using the primary completion loss while the encoder also receives gradients from the auxiliary loss. Each auxiliary loss is normalized using an exponential moving average with decay $\gamma_{\mathrm{EMA}} = 0.99$, and the adaptive weights w are updated only by the auxiliary loss with learning rate $\gamma_w = 5 \times 10^{-4}$. We include a zero-step term with weight $\beta_0 = 0.1$ and scale the auxiliary loss by $\lambda_{\mathrm{aux}} = 10.0$. The projection-based depth loss is applied only to the refinement module via a dedicated optimizer to prevent drift of the completion backbone. At test time, only encoder parameters are adapted with SGD ($\alpha_{\mathrm{TTA}} = 10^{-3}$, $K_{\mathrm{TTA}} = 3$ steps per frame), while the decoder, refinement module, and auxiliary modules are frozen and BatchNorm layers remain in evaluation mode.

C. Overall Results on Benchmarks

Table I reports quantitative comparison on ShapeNet, where PRISM achieves the lowest CD and competitive F-score among all baselines, demonstrating strong completion accuracy on synthetic data. On our industrial RoboArm-Depth dataset, PRISM—trained only on ShapeNet—also achieves strong reconstruction quality, with a CD of 67.74×10^{-4}, sub-

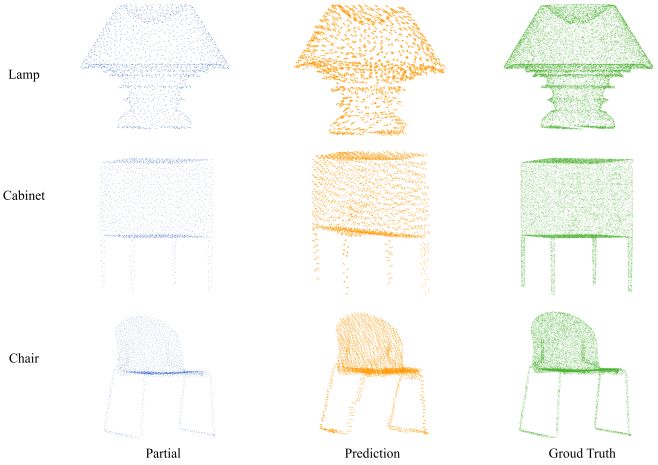

Fig. 4. Qualitative completion results on ShapeNet, showing the partial input, PRISM prediction, and ground-truth complete point cloud.

TABLE I
QUANTITATIVE COMPARISON ON SHAPENET. CD (L2) IS REPORTED AS $\times 10^{-4}$ AND F-SCORE (0–1). BASELINE NUMBERS ARE FROM THE ORIGINAL PAPERS OR OFFICIAL IMPLEMENTATIONS.

Method	CD ↓	F-score ↑
PCN	26.6	0.133
SeedFormer	9.2	0.472
PoinTr	10.9	0.464
CRA-PCN	8.5	–
PST-Net	8.7	0.475
HDANet	8.8	0.490
FSC	6.7	–
PCDreamer	8.95	**0.569**
IE-PMMA	7.0	0.480
PRISM (Ours)	**6.27**	0.497

stantially reducing reconstruction errors around self-occluded robot links and end-effectors that are critical for reliable robot envelope estimation in safety monitoring. This cross-domain generalization to real industrial LiDAR scenes highlights the effectiveness of PRISM's test-time adaptation mechanism under severe sparsity and occlusion. On ShapeNet, PRISM attains an EMD of 7.07×10^{-2} and an HD of 9.77×10^{-2}, indicating that the completed point clouds are geometrically faithful and densely aligned with the ground-truth surfaces.

D. Qualitative Results

Figure 4 and Figure 5 show qualitative completion results on ShapeNet and RoboArm-Depth, visualizing the input partial point cloud, the PRISM completion, and the ground-truth shapes. On ShapeNet, PRISM reconstructs more complete and sharper geometries under severe missing regions, while on RoboArm-Depth it recovers occluded links and end-effectors so that the completed envelopes closely match the ground-truth robot geometry from sparse industrial LiDAR.

E. Ablation Studies

We conduct ablation studies to analyze the contributions of meta-learning, geometry-aware auxiliary branches, and TTA.

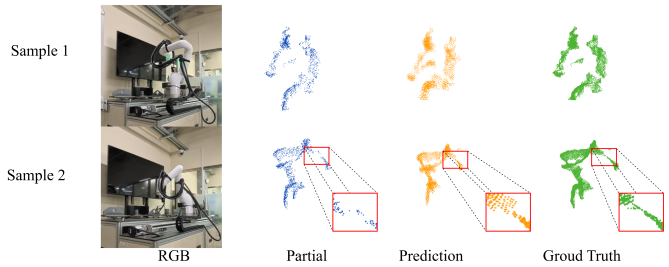

Fig. 5. Qualitative completion results on RoboArm-Depth, showing the input sparse LiDAR, PRISM prediction, and ground-truth complete point cloud.

TABLE II
EFFECT OF TEST-TIME ADAPTATION (TTA) ON SHAPENET. CD IS REPORTED AS $\times 10^{-4}$ AND F-SCORE AT THRESHOLD 0.01.

Method	TTA	CD ↓	F-score ↑
Meta only	✗	6.54	0.503
Meta+TTA (ours)	✓	6.27	0.497

Table II compares a meta-learned model without test-time adaptation (*Meta only*) against our full variant with encoder-only self-supervised adaptation (*Meta+TTA*) on ShapeNet. Meta-learned auxiliary weighting alone already yields strong performance, and enabling TTA further reduces CD from 6.54 to 6.27 $\times 10^{-4}$ at a modest trade-off in F-score (0.503 to 0.497), showing that depth-consistency driven adaptation can refine reconstruction while largely preserving surface coverage.

Table III evaluates auxiliary branches on RoboArm-Depth in a cross-domain setting, where the model is trained on ShapeNet and adapted to real LiDAR via TTA. Enabling 3D–depth alignment losses (L_{MM}+L_{OT}) already reduces CD from 101.11 to 72.16 $\times 10^{-4}$, and adding normal or denoising branches yields further gains, with the full configuration (*All aux*) achieving 67.74 $\times 10^{-4}$; this progressive improvement indicates that geometry-aware auxiliary supervision is crucial for transferring completion priors to sparse industrial LiDAR without retraining.

V. DISCUSSION AND CONCLUSION

We presented PRISM, a projection-regularized meta-learning framework for industrial LiDAR point cloud completion under severe sparsity and sensor-induced occlusions. Motivated by sparse observations from low-cost sensors, robot self-occlusion in collaborative workcells, and domain shifts from sensor reconfiguration, PRISM integrates projection-based depth supervision with episodic meta-learning to enable fast test-time adaptation without labels. Experiments on ShapeNet and our industrial RoboArm-Depth dataset demonstrate strong completion performance, with PRISM recovering occluded robot links and end-effectors critical for safety envelope estimation, while also revealing the remaining difficulty of strict cross-domain generalization from synthetic 360-degree views to single-view industrial scans. Ablation studies confirm that projection-based depth supervision and

TABLE III
ABLATION OF AUXILIARY BRANCHES ON ROBOARM-DEPTH. CD IS REPORTED AS $\times 10^{-4}$.

Method	Align	Normal	Denoise	CD ↓
No auxiliary	✗	✗	✗	101.11
Align only	✓	✗	✗	72.16
Align + Normal	✓	✓	✗	70.11
Align + Denoise	✓	✗	✓	70.43
All aux (ours)	✓	✓	✓	67.74

meta-learned auxiliary weighting each contribute significant performance gains under severe sparsity, indicating that geometric structure can still be effectively exploited even when source and target domains differ substantially.

From a deployment perspective, PRISM's label-free adaptation reduces manual annotation effort and supports deployment across heterogeneous manufacturing environments. Key challenges for broader adoption include extending to dynamic environments with moving robots via temporal modeling, improving robustness for specialized equipment geometries through meta-learning over equipment categories or CAD priors, and incorporating RGB cameras or other modalities to leverage vision-language models for zero-shot generalization. Addressing these aspects, particularly more reliable cross-domain zero-shot transfer to previously unseen industrial setups, is a natural direction for future work.

Industrial LiDAR point cloud completion under severe sparsity and occlusion is a critical enabler for safe automation in modern manufacturing. PRISM shows that projection-based depth supervision and meta-learned test-time adaptation can effectively address many of these challenges today, while also highlighting opportunities to further strengthen generalization in real-world deployment, providing a practical pathway toward scalable 3D perception for collaborative robot workcells, digital twins, and manufacturing systems.

REFERENCES

[1] Y. Lu, C. Liu, K. I.-K. Wang, H. Huang, and X. Xu, "Digital twin-driven smart manufacturing," *International Journal of Production Research*, vol. 58, no. 12, pp. 3767–3784, 2020.

[2] S. Zhai, S. Wu, D. Xu, and Y. Wang, "Point cloud-based deep learning in industrial production: A survey," *ACM Computing Surveys*, vol. 57, no. 1, pp. 1–38, 2024.

[3] A. Geiger, P. Lenz, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[4] H. Phan, W. Yang, and C. Xiao, "Domain generalization in 3d object detection from point cloud," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2022.

[5] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point completion network," in *2018 international conference on 3D vision (3DV)*. IEEE, 2018, pp. 728–737.

[6] X. Yu, Y. Rao, Z. Wang, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1588–1597.

[7] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, and D. Lin, "Density-aware chamfer distance as a comprehensive metric for point cloud completion," *arXiv preprint arXiv:2111.12702*, 2021.

[8] F. Lin, Y. Yue, S. Hou, X. Yu, Y. Xu, K. D. Yamada, and Z. Zhang, "Hyperbolic chamfer distance for point cloud completion," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 14 595–14 606.

[9] L. Jiang, R. Ma, L. Gu, Z. Wang, X. Zuo, and Y. Wang, "Pointmac: Meta-learned adaptation for robust test-time point cloud completion," *arXiv preprint arXiv:2510.10365*, 2025.

[10] J. Jiang, Q. Zhou, Y. Li, X. Zhao, M. Wang, L. Ma, J. Chang, J. Zhang, X. Lu *et al.*, "Pcotta: Continual test-time adaptation for multi-task point cloud understanding," *Advances in Neural Information Processing Systems*, vol. 37, pp. 96 229–96 253, 2024.

[11] A. Hatem, Y. Qian, and Y. Wang, "Point-tta: Test-time adaptation for point cloud registration using multitask meta-auxiliary learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 16 494–16 504.

[12] F. Langer, A. Milioto, A. Haag, J. Behley, and C. Stachniss, "Domain transfer for semantic segmentation of lidar data using deep neural networks," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 8263–8270.

[13] E. Magrini, F. Ferraguti, A. J. Ronga, F. Pini, A. De Luca, and F. Leali, "Human-robot coexistence and interaction in open industrial cells," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101846, 2020.

[14] C. Scholz, H.-L. Cao, E. Imrith, N. Roshandel, H. Firouzipouyaei, A. Burkiewicz, M. Amighi, S. Menet, D. W. Sisavath, A. Paolillo *et al.*, "Sensor-enabled safety systems for human-robot collaboration: A review," *IEEE Sensors Journal*, 2024.

[15] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[16] S. Xie, J. Gu, D. Gao, C. R. Qi, Y. Kawahara, and L. J. Guibas, "Unsupervised 3d learning from a single point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 676–11 684.

[17] P. Xiang, X. Wen, Y. Liu, Z. Han, M. Zwicker, and Y.-S. Liu, "Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8808–8817.

[18] J. Zou, Y. Tao, and Y. Yang, "Hierarchical distribution-aware network for point cloud completion," in *2025 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2025, pp. 1–8.

[19] Y. Wang, H. Zhu, and G. Wang, "Pst-net: Point cloud completion network based on local geometric feature reuse and neighboring recovery with taylor approximation," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–8.

[20] J. Zhang, X. Chen, L. Wang, and M. Liu, "Hspc-net: A hierarchical shape-preserving completion network for 3d point clouds," *PLoS ONE*, vol. 20, no. 8, p. e0287894, 2025.

[21] X. Wu, X. Wu, T. Luan, Y. Bai, Z. Lai, and J. Yuan, "Fsc: Few-point shape completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26 077–26 087.

[22] G. Wei, Y. Feng, L. Ma, C. Wang, Y. Zhou, and C. Li, "Pcdreamer: Point cloud completion through multi-view diffusion priors," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 27 243–27 253.

[23] R. Jia, J. Xue, S. Ma, W. Lu, and K. Wang, "Ie-pmma: point cloud completion through inverse edge-aware upsampling and precise multi-modal feature alignment," in *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025, pp. 1233–1241.

[24] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[25] D. Wang, E. Shelhamer, S. Liu, B. A. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," in *International Conference on Learning Representations*, 2021.

[26] A. Bartler, A. Bühler, F. Wiewel, M. Döbler, and B. Yang, "Mt3: Meta test-time training for self-supervised test-time adaption," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 3080–3090.

[27] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.