π^2: A Simple Framework for 2D-to-3D Registration

Hao Deng[1] **Xiangtai Yang**[1] **Guangmingzi Yang**[2] **Xijing Wang**[1]
Yuanxiao Ma[2] **Sisi Li**[1] **Zhiqiang Tian**[1] **Shaoyi Du**[1]*

[1]State Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University

[2]China Mobile Communications Group Co., Ltd.

denghao293@stu.xjtu.edu.cn dushaoyi@xjtu.edu.cn

Abstract

In this paper, we present π^2, a simple, efficient, and accurate image-to-point-cloud (I2P) registration framework. Instead of relying on intermediate targets such as 2D–3D correspondences, matching confidence, visibility masks, overlap regions, or pose-related proxies, we reformulate I2P registration as point-wise camera-centered coordinate regression. Given a target image and an unaligned LiDAR point cloud, π^2 directly predicts where each sampled 3D point should lie in the target camera frame. The predicted camera-centered point set is then aligned with the original point set through closed-form SVD-based rigid alignment, yielding the final relative pose without iterative pose optimization or a learned pose-regression head. To improve robustness under large viewpoint changes and sparse outdoor observations, we further introduce a lightweight camera-ray-based regularization objective that provides visibility-aware supervision during training, without requiring an additional visibility or overlap detection network. Experiments on KITTI and nuScenes show that π^2 outperforms the previous SoTA method ICL by 2.47 and 6.72 percentage points in registration success rate, respectively, while running in real time at about 25 FPS, corresponding to an up to 6$\times$ speedup over ICL. Project page: https://github.com/DengH293/P2I.

1 Introduction

I2P registration [6, 14] is a fundamental problem for 3D perception, aiming to estimate the relative rigid transformation between a camera image and a 3D point cloud map of the same scene [15]. It is particularly important in autonomous driving and mobile robotics, where accurate alignment between onboard camera observations and LiDAR maps is essential for visual localization [22, 9, 5], navigation [25], visual odometry [26], and map-based 3D reconstruction [16]. Despite its importance, outdoor I2P registration remains challenging due to the substantial modality gap between dense but depth-ambiguous images and sparse but metric LiDAR point clouds. In real-world driving scenes, this gap is further amplified by long-range sparsity, occlusion, repetitive urban structures, dynamic objects, illumination changes, and large viewpoint variations.

To bridge the modality gap between images and point clouds, existing learning-based I2P methods usually introduce cross-modal geometric cues before pose recovery. A representative line of work establishes 2D–3D associations between image pixels or patches and 3D points or local regions, and then estimates the camera pose using geometric solvers such as PnP, RANSAC, or differentiable variants [6, 19, 8, 11], as illustrated in the upper part of Fig. 1. Other methods avoid explicit point-to-pixel matching by predicting auxiliary geometric cues, such as frustum membership, overlapping regions, rays, or other pose-related quantities [4, 10]. Moreover, in outdoor map-based registration, the LiDAR map often covers a much larger spatial extent than the camera field of view. Therefore,

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

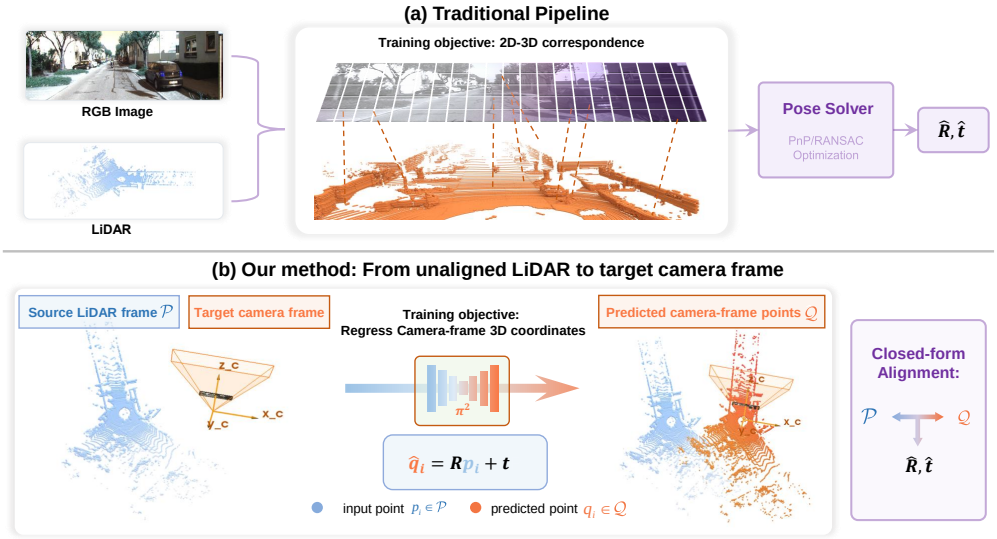

Figure 1: **From correspondence-driven I2P registration to camera-centered geometric regression.** (a) Representative I2P pipelines learn intermediate cross-modal cues, such as 2D–3D correspondences, and then recover the pose with a downstream solver. (b) Our method instead predicts the target-camera-frame coordinates of sampled LiDAR points conditioned on the image observation. The final pose is recovered by closed-form SVD alignment between the original sampled points and the predicted camera-frame points. This avoids treating explicit 2D–3D correspondence estimation as the supervised output and pose-recovery bottleneck.

many pipelines need to explicitly or implicitly identify the view-relevant subset of 3D geometry before reliable cross-modal fusion or pose recovery. Overall, existing methods typically construct a cross-modal geometric bridge first, and then infer the final pose from this bridge.

Despite their progress, the learning objectives of existing methods are still mainly defined on intermediate geometric representations, rather than directly on the final coordinate-frame alignment objective. This leads to two limitations in outdoor I2P registration. **First, from the efficiency perspective**, additional stages such as overlap or visibility prediction, candidate filtering, robust hypothesis verification, and iterative pose optimization may increase inference complexity. **Second, from the accuracy perspective**, the final pose is highly sensitive to the reliability of the preceding geometric cues. Ambiguous correspondences, inaccurate overlap estimation, or noisy pose proxies may propagate errors to subsequent cross-modal fusion, matching, or pose recovery. As a result, registration accuracy largely depends on whether these intermediate targets can provide sufficiently reliable and fine-grained geometric evidence for pose estimation. This raises a natural question: *can I2P registration be solved without learning correspondences or visibility proxies, by directly predicting the camera-frame coordinates of 3D points?*

Motivated by this observation, we reformulate outdoor I2P registration as point-wise camera-centered coordinate regression, as illustrated in Fig. 1(b). Rather than supervising cross-modal similarity, correspondences, visibility masks, or other pose-related proxies, we directly learn a pose-induced geometric target: the target-camera-frame coordinate of each sampled LiDAR point. We do not aim to remove cross-modal interaction; instead, we use it internally to infer where each sampled 3D point should lie in the target camera frame, while avoiding explicit correspondences as the supervised output and pose-recovery bottleneck.

To instantiate this formulation, we propose π^2, a camera-centered geometric regression framework with a direct output-and-pose-recovery pipeline. We denote our framework as π^2, a stylized form of P2I, since it maps sampled 3D points into the target camera-centered coordinate system conditioned on the image observation. Its simplicity lies in predicting source-point-indexed camera-frame coordinates and recovering the pose through closed-form SVD, without separate correspondence, overlap/visibility, or learned pose-prediction branches. Each sampled point retains its identity during cross-modal fusion and predicts its own target-camera-frame coordinate. We further introduce a training-time camera-ray-based auxiliary loss that encourages valid predicted coordinates to align

with their corresponding viewing rays. This loss injects camera-view geometric constraints at the objective level, stabilizing coordinate regression while preserving this direct coordinate-to-pose pipeline. Experiments on KITTI and nuScenes demonstrate that the proposed formulation is both accurate and efficient. Our method achieves the highest registration success rates on both benchmarks, reaching 99.96% on KITTI and 97.66% on nuScenes, improving over the previous baseline ICL by 2.47 and 6.72 percentage points, respectively. It also runs at 0.04s per image–point cloud pair, corresponding to real-time inference at about 25 FPS. These results support direct camera-frame coordinate regression as an efficient alternative to correspondence-driven I2P registration.

In summary, our contributions are threefold:

- We introduce a camera-centered geometric regression formulation for I2P registration, which directly predicts the target-camera-frame coordinates of sampled LiDAR points instead of treating 2D–3D correspondences or other intermediate geometric cues as the primary supervised output.
- We instantiate this formulation with a simple coordinate-to-pose pipeline that recovers the pose via closed-form SVD alignment, together with a training-time ray-consistency loss that imposes visibility-aware geometric constraints without additional visibility or overlap detection networks.
- Extensive experiments on KITTI and nuScenes demonstrate that our method achieves state-of-the-art registration success rates, runs in real time at about 25 FPS, and is up to 6$\times$ faster than recent baselines, while showing strong robustness to large initialization errors.

2 Related Work

Image-to-Point Cloud Registration. I2P registration estimates the relative pose between a camera image and a 3D point cloud. The modality gap between dense but depth-ambiguous images and sparse metric point clouds makes this task challenging, particularly under large viewpoint changes and partial overlap. Existing learning-based methods address this problem through different geometric representations. CMRNet [4] explores pose regression, while DeepI2P [10] predicts frustum membership and recovers the pose through inverse camera projection. Other approaches exploit virtual points [27], implicit correspondence learning [13], and ray-based geometric reasoning [12]. Our work studies a coordinate-regression formulation that uses cross-modal interaction to predict target-camera-frame coordinates for sampled LiDAR points, followed by closed-form rigid alignment.

Correspondence-Centric Registration. A major line of I2P research establishes 2D–3D correspondences and then estimates the pose using geometric solvers. Representative methods improve cross-modal association through sparse pixel-point matching [6, 21], dense correspondence learning [19], coarse-to-fine matching [8], transformer-based interaction [11], and graph reasoning [1]. The quality, inlier ratio, and spatial distribution of the estimated correspondences affect the reliability of downstream pose recovery. In outdoor I2P registration, sparse observations, occlusion, repetitive structures, and limited overlap can make these associations ambiguous. Visibility estimation, correspondence filtering, and robust hypothesis verification help address these difficulties, but can introduce additional prediction or processing stages. These considerations motivate predicting a camera-frame coordinate for each sampled source point, with the source–prediction pairing preserved by point identity. This removes the need to estimate an additional 2D–3D correspondence set for pose recovery, although pose accuracy depends on the geometric consistency of the predicted coordinates.

Coordinate Regression for Geometric Vision. Predicting 3D coordinates is an established approach to geometric localization and reconstruction. SCoRe forests [20] regress scene coordinates from RGB-D observations, associating image pixels with locations in a known scene coordinate frame for robust camera relocalization. DSAC [2] incorporates differentiable RANSAC into a learned localization pipeline, enabling training with an objective defined on the final pose. More recently, DUSt3R [24] predicts image-indexed pointmaps in a shared coordinate frame and integrates predictions from multiple image pairs through global alignment. Our method shares the principle of learning geometric quantities directly, while focusing on a source-point-anchored coordinate-to-pose interface for I2P registration. Given a target image and an unaligned metric LiDAR point cloud, π^2 predicts the target-camera-frame coordinate of each sampled source point. Unlike image-indexed scene coordinates or pointmaps, these predictions retain their pairing with the observed source points by construction, directly supporting closed-form 3D–3D Kabsch/SVD alignment. The contribution lies in this identity-preserving prediction and pose-recovery interface under sparse geometry and partial overlap, with a training-time camera-ray constraint providing complementary geometric supervision.

3 Methods

3.1 Preliminary: Intermediate-Target-Driven I2P Registration

Problem Formulation. Given an image $\mathcal{I}$ and a point cloud $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{N}$ of the same scene, image-to-point-cloud registration aims to estimate the rigid transformation $\mathcal{T} = (\mathbf{R}, \mathbf{t}) \in SE(3)$ that aligns the point-cloud coordinate frame to the target camera coordinate frame. Under $\mathcal{T}$, each point $\mathbf{p}_i$ is transformed into its camera-centered coordinate:

$$\mathbf{q}_i = \mathbf{R}\mathbf{p}_i + \mathbf{t}. \tag{1}$$

The goal of I2P registration is to recover $\mathcal{T}$ from the cross-modal observations $(\mathcal{I}, \mathcal{P})$.

Intermediate-Target-Driven Formulation. Due to the modality gap between images and point clouds, existing learning-based I2P methods usually introduce intermediate targets before recovering the final pose. This paradigm can be summarized as

$$(\mathcal{I}, \mathcal{P}) \xrightarrow{g_\theta} \mathcal{M}_\theta \xrightarrow{\mathcal{A}} \widehat{\mathcal{T}}, \tag{2}$$

where $\mathcal{M}_\theta$ denotes the predicted intermediate target, including 2D–3D matches, matching confidence, or feature similarity [6, 19, 8, 11], overlap or frustum cues [10, 19], or pose-related proxies [4]; and $\mathcal{A}$ denotes the subsequent pose recovery module, such as a geometric solver, an optimization procedure, or a learned pose head. In this view, existing methods first construct a cross-modal geometric bridge and then infer the final pose from it.

Limitations of intermediate targets. Although intermediate targets provide useful cross-modal cues, they are not the final quantity required by I2P registration.

1) **Indirect pose control.** Intermediate targets are optimized by surrogate losses, such as matching, confidence, visibility, or pose-proxy objectives. Their improvements do not necessarily reduce registration error, which also depends on the downstream solver or pose head.
2) **Output-level coupling.** Many methods explicitly output correspondences, overlap regions, visibility masks, or other cross-modal proxies. However, I2P registration ultimately requires a rigid transformation, so cross-modal interaction can be used internally rather than exposed as the primary supervised output.

These observations motivate a coordinate-level formulation that uses cross-modal interaction internally while directly predicting target-camera-frame coordinates for sampled 3D points, rather than supervising correspondences, visibility masks, overlap regions, or pose proxies.

3.2 I2P via Camera-Centered Geometric Regression

Reformulation and Pose Recovery. Building on the above observation, we formulate I2P registration as point-wise camera-centered coordinate regression. Let $\mathcal{P}_s = \{\mathbf{p}_i\}_{i=1}^{M}$ denote the input point set sampled from $\mathcal{P}$ by farthest point sampling (FPS). For each sampled point $\mathbf{p}_i$, its target is the camera-centered coordinate $\mathbf{q}_i$ defined in Eq. (1). Our network predicts this target as

$$\widehat{\mathbf{q}}_i = f_\theta(\mathcal{I}, \mathbf{p}_i, \mathcal{P}). \tag{3}$$

Once the predicted camera-centered point set $\widehat{\mathcal{Q}}_s = \{\widehat{\mathbf{q}}_i\}_{i=1}^{M}$ is obtained, the final pose is recovered by rigid alignment between the input points and their predicted camera-frame coordinates:

$$(\widehat{\mathbf{R}}, \widehat{\mathbf{t}}) = \arg\min_{\mathbf{R}\in SO(3),\, \mathbf{t}} \sum_{i=1}^{M} \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \widehat{\mathbf{q}}_i\|_2^2. \tag{4}$$

This is a standard Procrustes problem and can be solved in closed form via singular value decomposition (SVD). This formulation makes the network output directly compatible with 3D–3D rigid alignment, avoiding an additional correspondence construction or pose-regression head.

2D/3D Tokenization. To support camera-centered geometric regression, we transform both the point cloud and the image into geometry-aware token representations. For the point cloud, the token should preserve not only local 3D structure but also the metric identity of each sampled point. For the image, the token should encode both visual appearance and camera-view geometry. These modality-specific tokens provide a unified representation for subsequent cross-modal fusion.

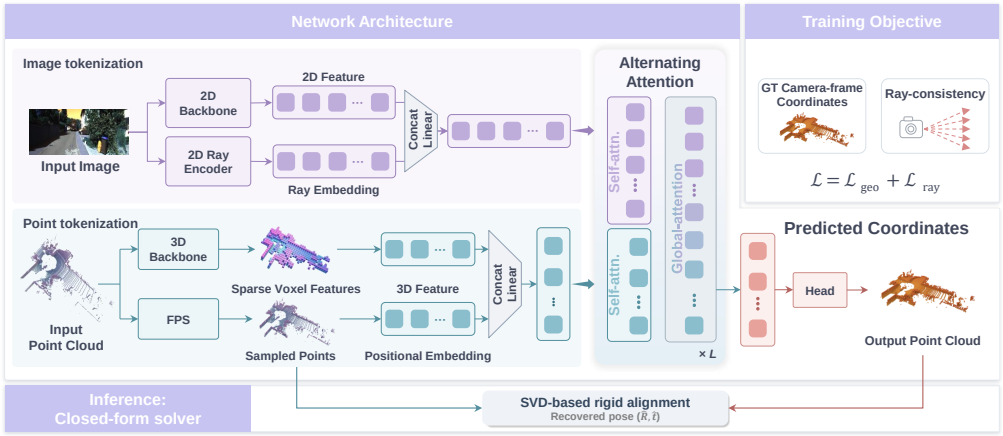

Figure 2: **Overview of the proposed π^2 framework**. The image and point-cloud branches produce geometry-aware tokens, which are fused by an alternating-attention Transformer. The fused point tokens regress camera-frame coordinates for sampled 3D points. During training, the model is supervised by coordinate regression and ray-consistency losses; during inference, the final pose is recovered by closed-form SVD-based rigid alignment.

(1) Point tokenization. We encode the voxelized point cloud using a sparse-convolution backbone, which is well suited to outdoor LiDAR data and efficiently aggregates local 3D context over sparse voxel neighborhoods. Specifically, the sparse encoder produces voxel-level features and associates each output voxel with its nearest input point:

$$\{(\widetilde{\mathbf{p}}_k, \widetilde{\mathbf{g}}_k^P)\}_{k=1}^{N_v} = \mathcal{E}_P(\mathcal{P}), \tag{5}$$

where $\widetilde{\mathbf{p}}_k$ is the representative input point of the k-th output voxel, and $\widetilde{\mathbf{g}}_k^P$ is the corresponding local 3D feature. We then apply FPS on the representative points to obtain the sampled point set $\mathcal{P}_s = \{\mathbf{p}_i\}_{i=1}^M$. Each sampled point $\mathbf{p}_i$ inherits the sparse feature of its selected representative point, denoted by $\mathbf{g}_i^P$. To preserve both relative geometry and global metric identity, we further encode each sampled point using both its absolute coordinate and its coordinate relative to the centroid of the sampled set with a lightweight MLP $\psi_P(\cdot)$. The final point token is constructed as

$$\mathbf{z}_i^P = \phi_P\Big(\Big[\mathbf{g}_i^P, \psi_P\Big(\big[(\mathbf{p}_i - \bar{\mathbf{p}})^\top, \mathbf{p}_i^\top\big]^\top\Big)\Big]\Big), \tag{6}$$

where $\bar{\mathbf{p}} = \frac{1}{M}\sum_{i=1}^M \mathbf{p}_i$ is the centroid of the point set.

(2) Image tokenization. For the image branch, we use a pretrained DINOv2 backbone [17] to extract token-aligned visual features $\{\mathbf{f}_j^I\}_{j=1}^{N_I}$, where $\mathbf{f}_j^I \in \mathbb{R}^C$ denotes the feature associated with the j-th image token. To encode camera-view layout, we associate each image token with its corresponding camera-ray direction. Let $\mathbf{u}_j \in \mathbb{R}^2$ denote the image-plane coordinate of the token center, and let $\bar{\mathbf{u}}_j = [\mathbf{u}_j^\top, 1]^\top$ be its homogeneous form. The final image token is constructed as

$$\mathbf{d}_j = \frac{\mathbf{K}^{-1}\bar{\mathbf{u}}_j}{\|\mathbf{K}^{-1}\bar{\mathbf{u}}_j\|_2}, \qquad \mathbf{z}_j^I = \phi_I\big([\mathbf{f}_j^I, \psi_r(\mathbf{d}_j)]\big), \tag{7}$$

where $\mathbf{K}$ is the camera intrinsic matrix, $\psi_r(\cdot)$ encodes the ray direction, and $\phi_I(\cdot)$ fuses the visual feature with the encoded ray cue. When the token resolution is lower than the input image resolution, we compute ray directions on the image grid and resize the dense ray map to the token grid.

Cross-Modal Geometry Fusion and Coordinate Regression. Given the point-token set $\mathcal{Z}^P = \{\mathbf{z}_i^P\}_{i=1}^M$ and the image-token set $\mathcal{Z}^I = \{\mathbf{z}_j^I\}_{j=1}^{N_I}$, we concatenate them into a unified multimodal sequence and process it with an alternating-attention Transformer:

$$\widetilde{\mathcal{Z}} = \Phi_\theta\big([\mathcal{Z}^P, \mathcal{Z}^I]\big). \tag{8}$$

Following the alternating-attention design, each block alternates between local and global attention to propagate cross-modal context at different scopes. The fused image tokens provide camera-view context for point-wise coordinate prediction, while the prediction target remains attached to sampled 3D points rather than explicit 2D–3D matches. Therefore, we retain only the fused point tokens $\widetilde{\mathcal{Z}}^P = \{\widetilde{\mathbf{z}}_i^P\}_{i=1}^M$ and decode each of them with a lightweight regression head:

$$\widehat{\mathbf{q}}_i = h_\theta(\widetilde{\mathbf{z}}_i^P). \tag{9}$$

The resulting camera-centered point set $\widehat{\mathcal{Q}} = \{\widehat{\mathbf{q}}_i\}_{i=1}^{M}$ is used to recover the final pose via Eq. (4).

Loss Function. We supervise camera-centered coordinate regression using coordinates normalized by a dataset-specific scale s. The geometric loss averages the squared errors over the M sampled points and their three coordinate components:

$$\mathcal{L}_{\text{geo}} = \frac{1}{3M} \sum_{i=1}^{M} \left\| \frac{\widehat{\mathbf{q}}_i}{s} - \frac{\mathbf{q}_i}{s} \right\|_2^2. \tag{10}$$

Here, $\mathbf{q}_i$ and $\widehat{\mathbf{q}}_i$ denote ground-truth and predicted coordinates in physical units. While $\mathcal{L}_{\text{geo}}$ directly supervises a pose-complete geometric target, coordinate regression remains challenging in outdoor scenes with large viewpoint changes, sparse observations, and many view-irrelevant 3D points. Existing methods often address this issue by introducing additional visibility, overlap, or frustum prediction branches to identify visually consistent geometry before pose recovery. Instead, we use a training-only ray-consistency constraint to inject camera-view geometry without adding any visibility prediction module at inference time.

Specifically, we associate sampled points with image rays by projecting their ground-truth camera-frame coordinates onto the image plane and matching valid projections to the nearest token-aligned camera rays within a distance threshold. Let $\mathcal{V}$ denote the resulting set of valid point-ray pairs, and let $\mathbf{r}_i \in \mathbb{R}^3$ be the camera ray associated with point i. The auxiliary ray loss is defined as

$$\mathcal{L}_{\text{ray}} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \left(1 - \frac{\widehat{\mathbf{q}}_i^{\top} \mathbf{r}_i}{\|\widehat{\mathbf{q}}_i\|_2 \, \|\mathbf{r}_i\|_2} \right). \tag{11}$$

This term encourages the prediction to lie not only near the target 3D coordinate, but also along the correct viewing direction. The final training objective is

$$\mathcal{L} = \mathcal{L}_{\text{geo}} + \lambda \mathcal{L}_{\text{ray}}. \tag{12}$$

Here, λ controls the relative weight of the ray loss. $\mathcal{L}_{\text{geo}}$ supervises the full camera-frame coordinates, while $\mathcal{L}_{\text{ray}}$ provides complementary viewing-direction supervision that is important for robust coordinate learning. The ray loss is a lightweight training-only camera-ray constraint: it requires no additional prediction branch and introduces no inference-time computation.

3.3 Analysis: From Coordinate Regression to Pose Recovery

We provide a local stability analysis to show why camera-centered coordinate regression is compatible with closed-form pose recovery. The purpose of this analysis is not to prove global learnability of the network, but to clarify that the supervised coordinate error directly perturbs the input of the SVD solver. For each sampled point $\mathbf{p}_i \in \mathcal{P}_s$, let $\mathbf{q}_i = \mathbf{R}\mathbf{p}_i + \mathbf{t}$ be its ground-truth camera-frame coordinate, and let $\mathbf{e}_i = \widehat{\mathbf{q}}_i - \mathbf{q}_i$ denote the prediction error. We define the root-mean-square coordinate error as

$$\delta = \left(\frac{1}{M} \sum_{i=1}^{M} \|\mathbf{e}_i\|_2^2 \right)^{1/2} = \sqrt{3}\, s \sqrt{\mathcal{L}_{\text{geo}}}. \tag{13}$$

Proposition 3.1 (Stability of coordinate-to-pose recovery)**.** *Let $(\widehat{\mathbf{R}}, \widehat{\mathbf{t}})$ be recovered by Eq. (4). If the sampled point set is non-degenerate,*

$$\lambda_{\min}(\Sigma_P) \geq \gamma > 0, \quad \Sigma_P = \frac{1}{M} \sum_{i=1}^{M} (\mathbf{p}_i - \bar{\mathbf{p}})(\mathbf{p}_i - \bar{\mathbf{p}})^{\top}, \tag{14}$$

then, for sufficiently small coordinate perturbations,

$$\|\widehat{\mathbf{R}} - \mathbf{R}\|_F + \|\widehat{\mathbf{t}} - \mathbf{t}\|_2 \leq C(\mathcal{P}_s)\delta, \tag{15}$$

where $C(\mathcal{P}_s)$ depends on the scale and conditioning of the sampled point set.

For a fixed normalization scale $s > 0$, Proposition 3.1 indicates that minimizing $\mathcal{L}_{\text{geo}}$ directly reduces the perturbation seen by the final SVD-based pose recovery step. In contrast, intermediate-target losses, such as correspondence scores, visibility masks, overlap labels, or pose proxies, affect the final pose only through additional matching, selection, or optimization stages. Thus, camera-frame coordinate regression provides a solver-compatible supervision signal that is directly tied to the final rigid alignment problem. The proof is provided in Appendix E.

Table 1: **Quantitative results on NuScenes and KITTI.** Acc. denotes the success ratio under RRE $< 5^\circ$ and RTE $< 2\,\mathrm{m}$. Baseline results are quoted from ICL [13].

Category	Method	*NuScenes*			*KITTI*		
		RTE(m)↓	RRE(°)↓	Acc(%)↑	RTE(m)↓	RRE(°)↓	Acc(%)↑
Matching-based	CorrI2P [19]	3.04±60.76	3.73±9.03	49.00	3.78±65.16	5.89±20.34	72.42
	VP2P-match [27]	0.89±1.44	2.15±7.03	88.33	0.75±1.13	3.29±7.99	83.04
	FreeReg [23]	-	-	-	0.95±1.05	2.06±3.21	91.68
	CoFiI2P [8]	1.21±9.55	2.54±8.93	-	0.31±0.20	1.24±0.84	-
	ICL [13]	**0.63±0.44**	2.13±3.75	90.94	**0.20±0.21**	1.24±2.34	97.49
Matching-free	Grid Cls. + PnP [10]	3.02±2.40	12.66±21.01	2.45	3.64±3.46	19.19±28.96	11.22
	DeepI2P(3D)[10]	2.88±2.12	20.65±12.24	2.26	4.06±3.54	24.73±31.69	3.77
	DeepI2P(2D)[10]	2.78±1.99	4.80±6.21	38.10	3.59±3.21	11.66±18.16	25.95
	Ours	0.95±0.49	**1.49±1.40**	**97.66**	0.38±0.19	**0.95±0.64**	**99.96**

4 Experiments

4.1 Datasets

Evaluation Datasets. We follow previous works [1, 27, 19] to conduct experiments on two widely adopted benchmarks, namely KITTI and NuScenes. In both datasets, 2D images and 3D point clouds are synchronously captured by cameras and LiDAR sensors.

KITTI Odometry [7]. Following the common protocol in previous work [10], sequences 00–08 are used for training, while sequences 09 and 10 are reserved for testing. To simulate calibration errors, we apply random perturbations consisting of translations within ± 10 m on the ground plane and arbitrary yaw rotations. The input image resolution is resized to 168×518, whose height and width are both multiples of 14 to be compatible with the DINO patch size.

nuScenes [3]. For the NuScenes dataset, image–point cloud pairs are generated using the official SDK, where the point cloud is aggregated from neighboring LiDAR frames and the image is taken from the current frame. We follow the official split of NuScenes and use 850 scenes for training and 150 scenes for testing. The perturbation strategy is consistent with that used on KITTI. Specifically, the input image resolution is resized to 168×322, whose height and width are both multiples of 14 to be compatible with the DINO patch size.

4.2 Registration Performance

Quantitative comparison. Table 1 summarizes the quantitative comparison on NuScenes and KITTI. Our method achieves the highest registration accuracy on both benchmarks, reaching 97.66% on NuScenes and 99.96% on KITTI. It also obtains the best rotation accuracy on both datasets, with RREs of 1.49° and 0.95°, respectively, while maintaining competitive translation errors. Compared with representative matching-free baselines, such as Grid Cls.+PnP and DeepI2P, our approach delivers substantial improvements across all metrics, demonstrating the advantage of directly regressing camera-centered coordinates rather than relying on proxy-based pose cues. Compared with recent matching-based methods, our method shows stronger final registration reliability, as indicated by its consistently higher accuracy on both benchmarks. Notably, compared with the strongest previous baseline ICL, our method is conceptually simpler, requiring neither an explicit overlapping-region detection module nor a learned pose-regression head. Despite this simpler formulation, it improves the registration accuracy by ***2.47*** percentage points on KITTI and ***6.72*** percentage points on NuScenes, while achieving real-time inference.

Qualitative comparison. Figure 3 presents qualitative comparisons on three KITTI examples under large initial misalignment. Starting from substantially perturbed inputs, our method consistently produces point-cloud projections that are visually closest to the ground-truth alignment, with clearer consistency along road layout, scene boundaries, and foreground structures. In contrast, VP2P-match often exhibits severe geometric drift or incorrect global alignment, while ICL achieves better recovery but still shows noticeable residual misalignment in challenging cases. Our method yields the lowest final errors in all three examples. These results further show that direct camera-centered coordinate regression leads to more reliable final registration than both correspondence-based and implicit relation modeling baselines.

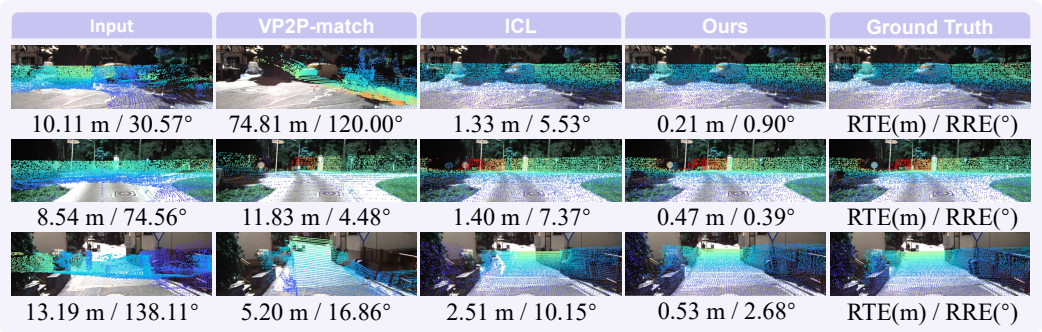

Figure 3: Qualitative comparison of I2P registration results on KITTI dataset.

Efficiency comparison. Table 2 compares model size and inference time. For baselines with available implementations, runtime is remeasured on the same RTX 3090 GPU used for π^2. The timing includes model inference and pose recovery, excluding data loading. For π^2, this covers feature extraction, multimodal fusion, coordinate regression, and SVD-based pose recovery. Our method achieves the fastest runtime of $\mathbf{0.04}$s per image–point cloud pair, corresponding to about 25 FPS. It is about $4.75\times$ faster than VP2P-match, $4.5\times$ faster than CoFiI2P, and $6\times$ faster than ICL. This efficiency benefits from our camera-centered coordinate regression formulation, which reduces pose recovery to closed-form rigid alignment without explicit correspondence construction, hypothesis verification, iterative refinement, or a learned pose-regression head.

Table 2: **Efficiency comparison.**

Method	Model size (MB)	Inference Time (s)
DeepI2P(3D) [10]	100.12	35.61
CorrI2P [19]	141.07	8.96
VP2P-match [27]	**30.73**	0.19
CoFiI2P [8]	39.09	0.18
ICL [13]	45.21	0.24
Ours	72.45	**0.04**

Robustness to large translation perturbations. In outdoor I2P registration, the standard evaluation protocol simulates initial calibration errors using translations within $\pm$10m on the ground plane, together with arbitrary yaw rotations around the vertical axis. However, practical outdoor registration may involve larger initial offsets than this standard setting. As shown in Fig. 4, ICL [13] suffers from increasing RTE/RRE and a clear drop in Acc as the translation range grows, indicating its sensitivity to large initial misalignment. In contrast, our method maintains lower pose errors and higher registration accuracy on both datasets, demonstrating stronger robustness to large-range initialization.

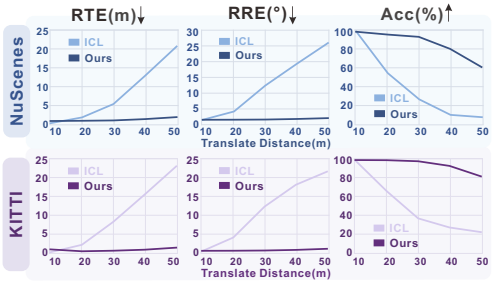

Figure 4: **Robustness to large translation perturbations.** Models are trained under the standard $\pm$10m translation perturbation and tested with progressively enlarged translation ranges up to 50m.

4.3 Ablation Study

Ablation on geometry-aware representations. Table 3 ablates relative point coordinates, absolute point coordinates, and ray-aware image embeddings. Using either relative or absolute coordinates already gives strong performance, and combining them further improves the results, showing that local geometry and global metric identity are complementary. Adding ray-aware image embeddings achieves the best performance on both NuScenes and KITTI, confirming the importance of camera-view geometry for camera-centered coordinate regression.

Ablation on sampled points and image resolution. Table 4 studies the trade-off between accuracy and runtime under different numbers of FPS-sampled points and image resolutions on KITTI. Increasing the number of sampled points improves accuracy up to $M = 1500$, while using $M = 2000$ brings negligible gains with higher computational cost. Similarly, increasing the image resolution improves accuracy, but the gain becomes marginal beyond 168×518. Based on this trade-off, we use $M = 1500$ and adopt a DINO-compatible image resolution of 168×518 for KITTI by default.

Table 3: **Ablation on geometry representations.** RelPos.: relative point position encoding $(\mathbf{p}_i - \bar{\mathbf{p}})$; AbsPos.: absolute point position encoding $\mathbf{p}_i$; RayEmb.: image ray embedding $\psi_r(\mathbf{d}_j)$. ✓: included, ✗: excluded.

RelPos.	AbsPos.	RayEmb.	*NuScenes*			*KITTI*		
			RTE(m)↓	RRE(°)↓	Acc(%)↑	RTE(m)↓	RRE(°)↓	Acc(%)↑
✓	✗	✗	**1.04±0.52**	**1.57±1.43**	**93.76**	0.62±0.54	**1.77±5.02**	**96.85**
✗	✓	✗	**1.01±0.55**	**1.54±1.49**	**94.55**	0.58±0.36	**1.68±6.61**	**97.44**
✓	✓	✗	**0.95±0.48**	**1.66±1.58**	**95.66**	0.41±0.28	**1.28±0.93**	**98.78**
✓	✓	✓	**0.95±0.49**	**1.49±1.40**	**97.66**	0.38±0.19	**0.95±0.64**	**99.96**

Table 4: **Ablation on point and image resolutions.**

Setting	Configuration	Acc.(%)↑	Time(ms)↓
Point	$M = 500$	97.65	31
	$M = 1000$	98.94	36
	$M = 1500$	99.96	42
	$M = 2000$	99.96	49
Image	126×392	98.66	38
	168×518	99.96	42
	210×644	99.96	55

Table 5: **Ablation on the pose recovery strategy.**

Variant	*NuScenes* Acc.(%)↑	*KITTI* Acc.(%)↑
w/o $\mathcal{L}_{ray}$	75.16	85.78
Learned pose head	89.85	94.11
RANSAC-SVD	97.48	99.79
Full model	97.66	99.96

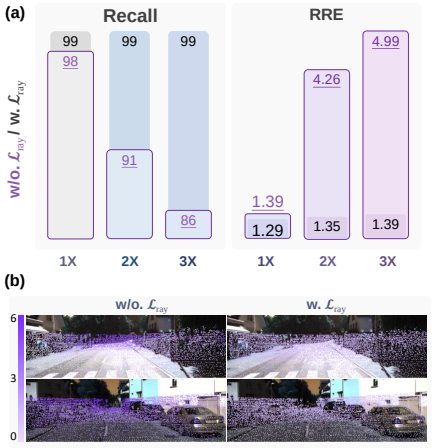

Figure 5: **Ablation on Ray-Consistency Loss.**

Ablation on Ray-Consistency Loss. Enlarging the input point-cloud scope introduces more out-of-view points and weakly related geometric structures, making effective geometry selection and cross-modal alignment more challenging. Instead of relying on additional visibility prediction or overlap-region estimation modules, we inject camera-view geometric constraints at the training-objective level through the $\mathcal{L}_{\text{ray}}$. Although $\mathcal{L}_{\text{geo}}$ already supervises full 3D coordinates, $\mathcal{L}_{\text{ray}}$ provides a complementary normalized direction constraint along camera rays, making the regression less sensitive to depth-scale variation and irrelevant out-of-view geometry. Fig. 5 analyzes this design from both quantitative and qualitative perspectives. In Fig. 5 (a), we enlarge the input point-cloud scope from $1\times$ to $3\times$. Without $\mathcal{L}_{\text{ray}}$, recall decreases and RRE increases as the scope grows, indicating reduced robustness to irrelevant 3D geometry. In contrast, the full model maintains more stable recall and rotation accuracy across different scopes. Fig. 5 (b) further shows that removing $\mathcal{L}_{\text{ray}}$ leads to larger projection misalignment around road boundaries and scene structures. These results demonstrate that $\mathcal{L}_{\text{ray}}$ improves camera-view geometric consistency and enhances robustness to large-scope point clouds without an explicit visibility or overlap prediction branch.

Ablation on objective and pose recovery strategy. Table 5 compares Ray-Consistency Loss and different pose recovery strategies. Removing Ray-Consistency Loss significantly degrades accuracy on both KITTI and NuScenes, indicating that it provides useful camera-view constraints for coordinate regression. Replacing closed-form SVD with a learned pose head also hurts performance, suggesting that point-wise camera-frame coordinates are better exploited by explicit rigid alignment than by global pose regression. RANSAC-SVD performs competitively but is slightly worse than direct SVD, likely because random sampling and thresholding may discard useful point-wise constraints and introduce hyperparameter sensitivity. These results support our final design: Ray-Consistency Loss for coordinate regression and deterministic SVD for pose recovery.

4.4 Generalization and Robustness

Full 6-DoF perturbations. We retrain and evaluate π^2 on KITTI with three-axis translations and unrestricted roll, pitch, and yaw. Under the same success criterion ($\text{RTE} < 2\,\text{m}$ and $\text{RRE} < 5°$), the model achieves 98.44% accuracy, compared with 99.96% under the standard horizontal-translation-and-yaw setting. The decrease of 1.52 percentage points indicates that π^2 maintains strong registration performance under full 6-DoF perturbations.

Table 6: **Image dependence.**

Image input	Acc. (%) ↑
Correctly paired	99.96
Intra-seq. shuffle	74.61
Cross-seq. shuffle	67.48

Table 7: **Indoor registration on 7-Scenes.** Recall (%).

Method	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Mean
2D3D-MATR	**96.9**	90.7	52.1	**95.5**	80.9	86.1	28.4	75.8
π^2	83.3	**95.5**	**80.4**	91.7	**93.2**	**93.5**	**39.1**	**82.4**

Dependence on the target image. We keep the trained model fixed and replace its KITTI test images with images from either the same sequence, at approximately 50m separation, or the other test sequence (09 versus 10). Table 6 shows substantial accuracy drops in both cases. Thus, although outdoor geometry provides useful priors, correctly paired images materially contribute to registration.

Few-shot cross-dataset adaptation. Starting from KITTI pretraining, we fine-tune on 5% of the nuScenes training pairs (3,146 pairs) for two epochs and evaluate on all 13,158 evaluation pairs. The model achieves 56.77% accuracy, with RTE 1.72 ± 1.33m and RRE $4.59 \pm 10.36°$. This experiment demonstrates partial transfer with limited target-domain supervision, while the remaining gap to dataset-specific training indicates a substantial domain shift.

Indoor registration. We evaluate on 7-Scenes using the split and Registration Recall protocol of 2D3D-MATR [11], with success defined by alignment RMSE below 10 cm. As shown in Table 7, π^2 achieves 82.4% mean recall, exceeding the reported 2D3D-MATR result by 6.6 percentage points and improving on five of seven scenes. These results extend the evaluation to indoor geometry, while the lower recall on Stairs indicates remaining challenges.

5 Conclusion

We presented π^2, a simple and effective matching-free framework for image-to-point cloud registration. Instead of relying on explicit 2D–3D correspondences, overlap prediction, or learned pose heads, our method directly regresses camera-centered coordinates for sampled 3D points and recovers the final pose through closed-form rigid alignment. With geometry-aware point and image representations and an auxiliary ray-consistency loss, the proposed framework achieves strong accuracy, robustness, and efficiency on both KITTI and nuScenes. These results highlight a formulation-level insight: rigid cross-modal registration can be effectively solved without explicit correspondence reasoning when the learning objective is defined on pose-complete geometric quantities. We hope this formulation can provide a useful basis for future research on scalable cross-modal registration.

Acknowledgments

This work was supported by New Generation Artificial Intelligence-National Science and Technology Major Project under Grant No. 2025ZD0124204, the National Natural Science Foundation of China under Grant Nos. U24A20252, 62327808, the Postdoctoral Research Funding Project of Shaanxi Province under Grant No. 2025BSHSDZZ023, the Fundamental Research Funds for the Central Universities under Grant No. xtr062025010 and xzy012026035, the Xi'an Jiaotong University-China Mobile Communications Group Co., Ltd. Joint Institute.

References

[1] Lin Bie, Shouan Pan, Siqi Li, Yining Zhao, and Yue Gao. Graphi2p: Image-to-point cloud registration with exploring pattern of correspondence via graph learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22161–22171, 2025. 3, 7

[2] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC—differentiable RANSAC for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 7

[4] Daniele Cattaneo, Matteo Vaghi, Augusto Luis Ballardini, Simone Fontana, Domenico G Sorrenti, and Wolfram Burgard. Cmrnet: Camera to lidar-map registration. In *2019 IEEE intelligent transportation systems conference (ITSC)*, pages 1283–1289. IEEE, 2019. 1, 3, 4

[5] Daniele Cattaneo and Abhinav Valada. Cmrnext: Camera to lidar matching in the wild for localization and extrinsic calibration. *IEEE Transactions on Robotics*, 2025. 1

[6] Mengdan Feng, Sixing Hu, Marcelo H Ang, and Gim Hee Lee. 2d3d-matchnet: Learning to match keypoints across 2d image and 3d point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4790–4796. IEEE, 2019. 1, 3, 4

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013. 7

[8] Shuhao Kang, Youqi Liao, Jianping Li, Fuxun Liang, Yuhao Li, Xianghong Zou, Fangning Li, Xieyuanli Chen, Zhen Dong, and Bisheng Yang. Cofii2p: Coarse-to-fine correspondences for image-to-point cloud registration. *arXiv preprint arXiv:2309.14660*, 2023. 1, 3, 4, 7, 8

[9] Minjung Kim, Junseo Koo, and Gunhee Kim. Ep2p-loc: End-to-end 3d point to 2d pixel localization for large-scale visual localization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 21527–21537, 2023. 1

[10] Jiaxin Li and Gim Hee Lee. Deepi2p: Image-to-point cloud registration via deep classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15960–15969, 2021. 1, 3, 4, 7, 8

[11] Minhao Li, Zheng Qin, Zhirui Gao, Renjiao Yi, Chenyang Zhu, Yulan Guo, and Kai Xu. 2d3d-matr: 2d-3d matching transformer for detection-free registration between images and point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14128–14138, 2023. 1, 3, 4, 10

[12] Xinjun Li, Wenfei Yang, Zhixin Cheng, Jiacheng Deng, Fei Wang, Chen Qian, and Tianzhu Zhang. Rayi2p: Learning rays for image-to-point cloud registration. In *The Fourteenth International Conference on Learning Representations*, 2024. 3

[13] Xinjun Li, Wenfei Yang, Jiacheng Deng, Zhixin Cheng, Xu Zhou, and Tianzhu Zhang. Implicit correspondence learning for image-to-point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16922–16931, 2025. 3, 7, 8, 14, 15, 16

[14] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3d point clouds. In *European conference on computer vision*, pages 15–29. Springer, 2012. 1

[15] Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua. Accurate non-iterative o (n) solution to the pnp problem. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. Ieee, 2007. 1

[16] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16695–16705, 2025. 1

[17] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 5

[18] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 14

[19] Siyu Ren, Yiming Zeng, Junhui Hou, and Xiaodong Chen. Corri2p: Deep image-to-point cloud registration via dense correspondence. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(3):1198–1208, 2022. 1, 3, 4, 7, 8

[20] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013. 3

[21] Bing Wang, Changhao Chen, Zhaopeng Cui, Jie Qin, Chris Xiaoxuan Lu, Zhengdi Yu, Peijun Zhao, Zhen Dong, Fan Zhu, Niki Trigoni, et al. P2-net: Joint description and detection of local features for pixel and point matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16004–16013, 2021. 3

[22] Guangming Wang, Yu Zheng, Yuxuan Wu, Yanfeng Guo, Zhe Liu, Yixiang Zhu, Wolfram Burgard, and Hesheng Wang. End-to-end 2d-3d registration between image and lidar point cloud for vehicle localization. *IEEE Transactions on Robotics*, 2025. 1

[23] Haiping Wang, Yuan Liu, Bing Wang, Yujing Sun, Zhen Dong, Wenping Wang, and Bisheng Yang. Freereg: Image-to-point cloud registration leveraging pretrained diffusion models and monocular depth estimators. *arXiv preprint arXiv:2310.03420*, 2023. 7

[24] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20697–20709, 2024. 3

[25] Fan Yang, Chen Wang, Cesar Cadena, and Marco Hutter. iPlanner: Imperative Path Planning. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. 1

[26] Chunran Zheng, Wei Xu, Zuhao Zou, Tong Hua, Chongjian Yuan, Dongjiao He, Bingyang Zhou, Zheng Liu, Jiarong Lin, Fangcheng Zhu, et al. Fast-livo2: Fast, direct lidar–inertial–visual odometry. *IEEE Transactions on Robotics*, 41:326–346, 2024. 1

[27] Junsheng Zhou, Baorui Ma, Wenyuan Zhang, Yi Fang, Yu-Shen Liu, and Zhizhong Han. Differentiable registration of images and lidar point clouds with voxelpoint-to-pixel matching. *Advances in Neural Information Processing Systems*, 36:51166–51177, 2023. 3, 7, 8

Appendix

For a comprehensive understanding of our π^2, we provide additional implementation details, evaluation protocols, experimental analyses, limitations, and theoretical proof in this appendix. The table of contents below provides a clear overview of the supplementary material.

Contents

A. Implementation Details 13
A.1. Architecture Details 13
A.2. Training and Inference Details 14
B. Evaluation Protocol and Baseline Details 14
B.1. Dataset and Perturbation Protocol 14
B.2. Evaluation Metrics 15
C. Additional Experimental Results 15
C.1. Sensitivity to Camera Intrinsics 15
C.2. Sensitivity to the Ray-Loss Weight 15
C.3. Matched-Backbone Comparison 16
C.4. Additional Qualitative Results 16
D. Limitations and Future Work 17
E. Proof of Proposition 3.1 18

A Implementation Details

A.1 Architecture Details

We use the same overall architecture for KITTI and nuScenes. The model follows a two-branch tokenization design followed by multimodal Transformer fusion and point-wise coordinate regression.

For the point-cloud branch, each input point is represented by its normalized 3D coordinate, LiDAR intensity, and surface normal, resulting in a 7-dimensional point feature. The point cloud is voxelized before sparse convolution. For KITTI, we use a physical voxel size of 0.1m and a point-cloud scale of 150, corresponding to a normalized voxel size of $0.1/150$. For nuScenes, we use a physical voxel size of 0.3m and a point-cloud scale of 50, corresponding to a normalized voxel size of $0.3/50$. A sparse downsampling encoder with channel dimensions $16, 32, 64, 128, 256$ and four stride-2 stages is used to aggregate local 3D geometry. The resulting sparse feature is concatenated with a lightweight point feature embedding and projected to a 512-dimensional point token.

We then apply farthest point sampling and keep at most $M = 1500$ point tokens for each image–point-cloud pair. To preserve the metric identity of sampled points, each sampled point is further encoded by a 6-dimensional positional descriptor, formed by concatenating its relative coordinate and absolute coordinate. This descriptor is passed through an MLP with output dimension 128 and fused with the 512-dimensional point token.

For the image branch, we use a pretrained DINOv2 ViT-S/14 backbone to extract patch-level image tokens. For KITTI, the image resolution is 168×518, which produces a 12×37 token grid. For nuScenes, the image resolution is 168×322, which produces a 12×23 token grid. To encode camera-view geometry, we compute a dense camera-ray map from the intrinsic matrix and resize it to the DINO token resolution. Each token-aligned ray direction is encoded by a two-layer MLP with dimensions $3 \to 64 \to 128$ and fused with the visual token, yielding a 512-dimensional ray-aware image token.

The sampled point tokens and image tokens are concatenated into a single multimodal sequence and processed by a 12-layer alternating-attention Transformer with 8 attention heads and hidden dimension 512. After cross-modal fusion, only the point tokens are retained. A three-layer MLP regression head maps each fused point token to a 3D coordinate in the target camera frame. The final model does not use an explicit 2D–3D correspondence head, overlap prediction branch, visibility segmentation module, RANSAC module, or learned pose-regression head.

A.2 Training and Inference Details

Experimental environment. We implement π^2 in PyTorch [18] with Python 3.10.18, PyTorch 2.5.0, CUDA 12.4, and cuDNN 9.1.0. All experiments are conducted on a Linux workstation equipped with four NVIDIA GeForce RTX 3090 GPUs and two Intel Xeon Silver 4314 CPUs. Unless otherwise specified, training uses two RTX 3090 GPUs with a batch size of 12 per GPU, resulting in a total batch size of 24. We train the model using point-wise camera-centered coordinate supervision together with the auxiliary ray-consistency loss.

Training setting. For KITTI, we follow the standard split and train on sequences 00–08 while evaluating on sequences 09–10. The input image resolution is 168×518. We use a point-cloud scale of 150 and a physical voxel size of 0.1m. During training and evaluation, we simulate initial registration errors by perturbing the point cloud with translations sampled within ± 10m on the ground plane and arbitrary yaw rotations. For nuScenes, we follow the official train/validation split. The input image resolution is 168×322. We use a point-cloud scale of 50 and a physical voxel size of 0.3m. The same perturbation protocol is used: translations are sampled within ± 10m on the ground plane, and yaw rotations are sampled over the full 2π range. Unless otherwise specified, models are trained for 100 epochs with automatic mixed precision and gradient clipping with maximum norm 1.0. We use the AdamW optimizer with learning rate 1×10^{-4} and weight decay 10^{-6}. The learning-rate scheduler uses no warmup and decays to a minimum learning-rate ratio of 0.01.

Inference details. During inference, the network predicts camera-centered coordinates for the sampled 3D points in a single forward pass. The final rigid transformation is recovered by solving a 3D–3D Procrustes alignment problem between the sampled input points and their predicted camera-frame coordinates. We use the closed-form Kabsch/SVD solution for this step. This deterministic pose recovery avoids RANSAC, PnP, iterative refinement, and learned pose heads, keeping inference simple and efficient.

B Evaluation Protocol and Baseline Details

B.1 Dataset and Perturbation Protocol

This subsection provides additional details of the data construction and perturbation protocol used in our experiments. The dataset splits and basic preprocessing settings are described in Sec. 4; here we focus on the exact construction of evaluation pairs and initial pose perturbations.

For each image–point cloud pair, we use the calibrated LiDAR-to-camera transformation as the ground-truth transformation $\mathcal{T} = (\mathbf{R}, \mathbf{t}) \in \mathrm{SE}(3)$. To simulate an initially miscalibrated input, we apply a random perturbation $\Delta\mathcal{T} = (\Delta\mathbf{R}, \Delta\mathbf{t})$ to the point cloud before registration. The network takes the target image and the perturbed point cloud as input, and is trained to recover the relative transformation that aligns the perturbed LiDAR point cloud to the target camera coordinate frame.

The translational perturbation is sampled on the horizontal ground plane:

$$\Delta\mathbf{t} = [\Delta x,\ 0,\ \Delta z]^\top, \quad \Delta x, \Delta z \sim \mathcal{U}(-10, 10), \tag{16}$$

where the vertical component is kept unchanged. The rotational perturbation is sampled as a yaw rotation around the vertical axis:

$$\Delta\mathbf{R} = \mathbf{R}_y(\Delta\theta), \quad \Delta\theta \sim \mathcal{U}(-\pi, \pi). \tag{17}$$

Here $\mathbf{R}_y(\cdot)$ denotes rotation around the vertical axis. This perturbation setting follows the outdoor driving scenario, where the dominant initialization errors usually come from horizontal localization drift and heading uncertainty.

Given the ground-truth transformation $\mathcal{T}$ and the perturbation $\Delta\mathcal{T}$, the perturbed input point cloud is generated by applying the inverse perturbed pose to the target-frame point cloud. Equivalently, the model is required to estimate the residual rigid transformation from the perturbed LiDAR frame to the target camera frame. For our method, training and testing use the same perturbation range unless otherwise specified. The comparison in Table 1 follows the same public dataset splits, perturbation ranges, and evaluation metrics. Baseline results are quoted from the published ICL paper [13], while results for π^2 are obtained under the settings described above. This comparison does not imply identical preprocessing or identical randomly sampled perturbation instances.

B.2 Evaluation Metrics

To quantitatively evaluate image-to-point-cloud registration, we report three metrics: relative translation error (RTE), relative rotation error (RRE), and registration accuracy (Acc.). Let $(\mathbf{R}, \mathbf{t})$ denote the ground-truth rigid transformation from the LiDAR coordinate frame to the target camera coordinate frame, and let $(\widehat{\mathbf{R}}, \widehat{\mathbf{t}})$ denote the estimated transformation.

Relative Translation Error. The relative translation error measures the Euclidean distance between the estimated translation vector and the ground-truth translation vector:

$$\mathrm{RTE} = \left\| \widehat{\mathbf{t}} - \mathbf{t} \right\|_2. \tag{18}$$

RTE is reported in meters. A lower RTE indicates a more accurate translation estimate.

Relative Rotation Error. The relative rotation error measures the rotational discrepancy between the estimated rotation and the ground-truth rotation. We first compute the residual rotation matrix between the estimated and ground-truth transformations:

$$\mathbf{R}_{\mathrm{rel}} = \widehat{\mathbf{R}}^{-1}\mathbf{R}. \tag{19}$$

We then convert the residual rotation matrix $\mathbf{R}_{\mathrm{rel}}$ to Euler angles $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \gamma_3]^\top$, and define RRE as the sum of absolute angular deviations along the three axes:

$$\mathrm{RRE} = \sum_{i=1}^{3} |\gamma_i|. \tag{20}$$

RRE is reported in degrees, and a lower RRE indicates a more accurate rotation estimate.

Registration Accuracy. Following previous I2P registration works [13], we further report the registration accuracy (Acc.) as the percentage of successfully registered image–point cloud pairs. A registration result is considered successful only when both the rotation and translation errors are below predefined thresholds:

$$\mathrm{Acc.} = \frac{1}{N}\sum_{i=1}^{N}[\mathrm{RRE}_i < 5^\circ \wedge \mathrm{RTE}_i < 2\mathrm{m}]. \tag{21}$$

Here, $[\cdot]$ equals 1 if the condition inside the bracket is satisfied and 0 otherwise. Acc. is reported as a percentage, and a higher value indicates better registration reliability.

C Additional Experimental Results

C.1 Sensitivity to Camera Intrinsics

We keep the trained model fixed and perturb only the test-time intrinsics used to construct image-ray embeddings. For each pair, a shared relative focal-length error $\epsilon_f \sim \mathcal{U}(-a, a)$ scales both f_x and f_y, while independent principal-point offsets $\epsilon_x, \epsilon_y \sim \mathcal{U}(-b, b)$ shift c_x and c_y in pixels. RGB images, input point clouds, ground-truth poses, and SVD pose recovery remain unchanged.

Table 8: **Joint test-time intrinsic perturbations on KITTI.** RTE/RRE are per-pair means $\pm$ population standard deviations, including failures.

Focal / principal point	RTE (m) ↓	RRE (°) ↓
±10% / ±10 px	0.40 ± 0.21	1.10 ± 0.72
±20% / ±20 px	0.43 ± 0.22	1.41 ± 0.87
±30% / ±30 px	0.47 ± 0.26	1.78 ± 1.16
±40% / ±40 px	0.52 ± 0.29	2.14 ± 1.33

C.2 Sensitivity to the Ray-Loss Weight

Table 9 examines the weight λ in $\mathcal{L} = \mathcal{L}_{\mathrm{geo}} + \lambda\mathcal{L}_{\mathrm{ray}}$ on KITTI, with coordinate normalization as defined in the main paper. The reported sweep favors $\lambda = 0.25$, while $\lambda = 0.1$ gives similar mean pose errors. Larger weights increase both translation and rotation errors, indicating that directional supervision should complement the full coordinate objective rather than dominate it.

Table 9: **Ray-loss weight sensitivity on KITTI.** RTE/RRE are mean $\pm$ standard deviation.

λ	RTE (m) $\downarrow$	RRE ($^\circ$) $\downarrow$
0.1	0.39 ± 0.21	0.96 ± 0.77
0.25	0.38 ± 0.19	0.95 ± 0.64
0.5	0.45 ± 0.30	1.54 ± 1.22
1.0	0.48 ± 0.31	1.84 ± 1.34

C.3 Matched-Backbone Comparison

To assess whether the observed advantage depends only on the image backbone, we compare π^2 and ICL [13] with ResNet and DINOv2 image encoders on KITTI. The ResNet ICL result is quoted from [13]; DINOv2 ICL is our reimplementation under the same public split, perturbation ranges, and evaluation metrics. Table 10 shows that π^2 improves accuracy by 1.60 and 2.06 percentage points with ResNet and DINOv2, respectively. These comparisons support gains with either backbone family, while leaving other architectural and optimization differences between the methods intact.

Table 10: **Matched image-backbone comparison on KITTI.** Acc. (%) uses the standard 2m/5° success criterion.

Image backbone	ICL	π^2	Gain (points)
ResNet	97.49	99.09	+1.60
DINOv2	97.90	99.96	+2.06

C.4 Additional Qualitative Results

Visualization of coordinate transformation. In the main paper, we compare the registration results of our method with those of existing approaches. To further understand the behavior of our coordinate-regression formulation, Figure 6 visualizes the intermediate camera-centered coordinate regression results and the final pose recovery process. Different from matching-based methods that first establish 2D–3D correspondences, our method directly predicts the 3D coordinate of each sampled input point in the target camera coordinate frame. As shown in the successful cases, the direct model output is already close to the ground-truth camera-frame point cloud, which provides a favorable geometric basis for the subsequent closed-form SVD alignment. The predicted point set preserves a similar global structure to the ground truth, enabling accurate recovery of the relative pose.

The last row shows a representative failure case. Although the registration fails, the model still largely preserves the rigid shape of the point cloud. However, the predicted coordinates are only partially consistent with the target camera-frame geometry and exhibit local geometric distortion, which eventually leads to inaccurate pose recovery. This visualization provides an intuitive explanation of how camera-centered coordinate regression supports closed-form pose recovery, while also showing that the subsequent rigid alignment may fail when the predicted coordinate set is geometrically distorted or only partially aligned with the target frame.

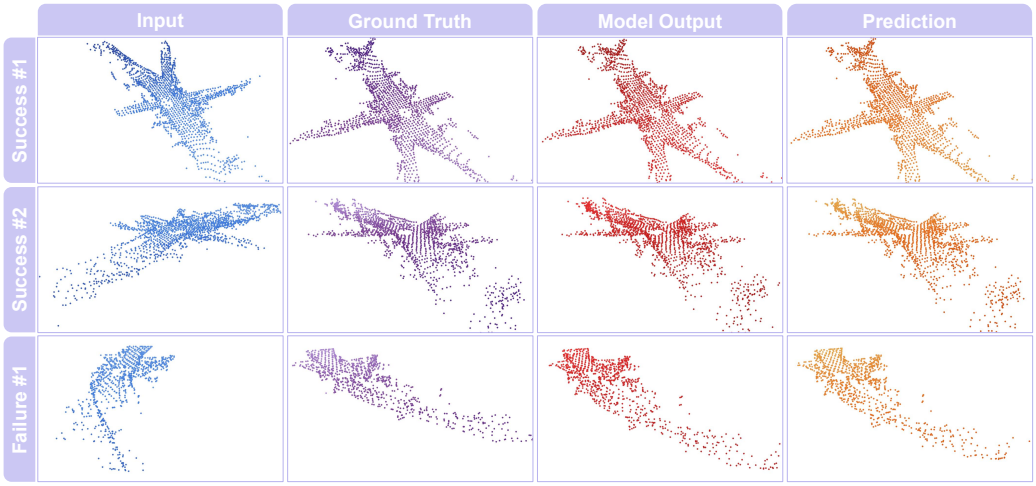

Figure 6: **Visualization of camera-centered coordinate regression and pose recovery.** We visualize the input point cloud, ground-truth camera-frame point cloud, model output, and the final prediction after SVD-based rigid alignment. The first two rows show successful registration cases, where the predicted camera-frame coordinates are geometrically consistent with the ground truth and lead to accurate pose recovery. The last row shows a representative failure case, where partially distorted coordinate predictions result in inaccurate final alignment.

D Limitations and Future Work

Limitations. Although π^2 achieves strong registration accuracy and real-time inference speed, several limitations remain. As shown in the failure case in Figure 6, the model can still preserve the overall rigid shape of the input point cloud, but the predicted camera-centered coordinates may be only partially consistent with the target camera-frame geometry. In this case, local geometric distortion or biased global placement in the predicted coordinate set can lead the subsequent SVD-based rigid alignment to recover an inaccurate pose. This suggests that reliable pose recovery requires not only preserving the global shape of the input point cloud, but also placing the predicted coordinates correctly in the target camera frame. Therefore, the performance of our method is still limited by the geometric consistency of the regressed camera-frame coordinates.

Future work. Section 4.4 extends our evaluation to full 6-DoF perturbations, indoor scenes, and few-shot cross-dataset adaptation. Nevertheless, these experiments do not establish unrestricted cross-sensor or zero-shot generalization, and the adaptation result still leaves a substantial performance gap under domain shift. Future work will explore more diverse sensor configurations, handheld sensors, and scene types, as well as robustness to inaccurate training intrinsics. It would also be valuable to introduce uncertainty estimation, confidence-aware rigid alignment, or lightweight visibility-aware weighting to reduce the influence of unreliable coordinate predictions while preserving the simplicity and efficiency.

Broader Impacts

This work studies image-to-point cloud registration, a core component for cross-modal localization and 3D perception. Its potential positive societal impacts include improving the robustness and efficiency of camera-LiDAR localization, navigation, visual odometry, and map-based 3D perception systems, which may benefit autonomous driving, mobile robotics, and other embodied perception applications.

Potential negative impacts mainly arise from safety-critical deployment and privacy-sensitive uses. If the registration result is incorrect under adverse weather, domain shift, sensor degradation, low-overlap observations, or unusual viewpoints, downstream perception or localization systems may fail, which could be harmful in autonomous driving or robotic navigation. In addition, more accurate cross-modal localization could potentially be misused for surveillance or spatial tracking applications.

Therefore, this method should not be deployed in safety-critical settings without extensive validation, uncertainty monitoring, failure detection, and system-level safeguards.

E Proof of Proposition 3.1

We provide the proof of Proposition 3.1. Let

$$\bar{\mathbf{p}} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{p}_i, \qquad \bar{\mathbf{q}} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{q}_i, \qquad \bar{\widehat{\mathbf{q}}} = \frac{1}{M}\sum_{i=1}^{M}\widehat{\mathbf{q}}_i.$$

Since $\mathbf{q}_i = \mathbf{R}\mathbf{p}_i + \mathbf{t}$ and $\widehat{\mathbf{q}}_i = \mathbf{q}_i + \mathbf{e}_i$, we have

$$\bar{\mathbf{q}} = \mathbf{R}\bar{\mathbf{p}} + \mathbf{t}, \qquad \bar{\widehat{\mathbf{q}}} = \bar{\mathbf{q}} + \bar{\mathbf{e}}, \qquad \bar{\mathbf{e}} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{e}_i.$$

After centering, define

$$\mathbf{x}_i = \mathbf{p}_i - \bar{\mathbf{p}}, \qquad \widehat{\mathbf{y}}_i = \widehat{\mathbf{q}}_i - \bar{\widehat{\mathbf{q}}}, \qquad \widetilde{\mathbf{e}}_i = \mathbf{e}_i - \bar{\mathbf{e}}.$$

Then

$$\widehat{\mathbf{y}}_i = \mathbf{R}\mathbf{x}_i + \widetilde{\mathbf{e}}_i.$$

The closed-form rigid alignment estimates the rotation from the cross-covariance matrix

$$\widehat{\mathbf{H}} = \frac{1}{M}\sum_{i=1}^{M}\widehat{\mathbf{y}}_i\mathbf{x}_i^\top.$$

Substituting $\widehat{\mathbf{y}}_i = \mathbf{R}\mathbf{x}_i + \widetilde{\mathbf{e}}_i$, we obtain

$$\widehat{\mathbf{H}} = \mathbf{R}\Sigma_P + \Delta, \qquad \Delta = \frac{1}{M}\sum_{i=1}^{M}\widetilde{\mathbf{e}}_i\mathbf{x}_i^\top, \tag{22}$$

where

$$\Sigma_P = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i\mathbf{x}_i^\top.$$

In the noiseless case, i.e., $\Delta = \mathbf{0}$, the cross-covariance is $\mathbf{R}\Sigma_P$. Since Σ_P is symmetric positive definite under the non-degeneracy condition $\lambda_{\min}(\Sigma_P) \geq \gamma > 0$, the polar/SVD factor of $\mathbf{R}\Sigma_P$ recovers the ground-truth rotation $\mathbf{R}$.

We next bound the perturbation term Δ. By Cauchy–Schwarz,

$$\|\Delta\|_F \leq \left(\frac{1}{M}\sum_{i=1}^{M}\|\mathbf{x}_i\|_2^2\right)^{1/2}\left(\frac{1}{M}\sum_{i=1}^{M}\|\widetilde{\mathbf{e}}_i\|_2^2\right)^{1/2}. \tag{23}$$

Since centering does not increase the total squared error,

$$\frac{1}{M}\sum_{i=1}^{M}\|\widetilde{\mathbf{e}}_i\|_2^2 \leq \frac{1}{M}\sum_{i=1}^{M}\|\mathbf{e}_i\|_2^2 = \delta^2,$$

we have

$$\|\Delta\|_F \leq s_P\delta, \qquad s_P = \left(\frac{1}{M}\sum_{i=1}^{M}\|\mathbf{p}_i - \bar{\mathbf{p}}\|_2^2\right)^{1/2}. \tag{24}$$

For sufficiently small perturbations, the polar/SVD factor is locally Lipschitz around $\mathbf{R}\Sigma_P$. Standard perturbation stability of the polar decomposition gives

$$\|\widehat{\mathbf{R}} - \mathbf{R}\|_F \leq C_R(\mathcal{P}_s)\|\Delta\|_F \leq C_R(\mathcal{P}_s)s_P\delta, \tag{25}$$

where $C_R(\mathcal{P}_s)$ depends on the conditioning of Σ_P, in particular on the lower bound γ.

The translation is recovered from the centroids:

$$\widehat{\mathbf{t}} = \bar{\widehat{\mathbf{q}}} - \widehat{\mathbf{R}}\bar{\mathbf{p}}, \qquad \mathbf{t} = \bar{\mathbf{q}} - \mathbf{R}\bar{\mathbf{p}}.$$

Therefore,

$$\begin{aligned} \|\widehat{\mathbf{t}} - \mathbf{t}\|_2 &= \left\| \bar{\widehat{\mathbf{q}}} - \bar{\mathbf{q}} - (\widehat{\mathbf{R}} - \mathbf{R})\bar{\mathbf{p}} \right\|_2 \\ &\leq \|\bar{\mathbf{e}}\|_2 + \|\widehat{\mathbf{R}} - \mathbf{R}\|_F \|\bar{\mathbf{p}}\|_2. \end{aligned} \tag{26}$$

Since $\|\bar{\mathbf{e}}\|_2 \leq \delta$, combining Eq. (25) and Eq. (26) yields

$$\|\widehat{\mathbf{t}} - \mathbf{t}\|_2 \leq C_t(\mathcal{P}_s)\delta. \tag{27}$$

Finally, combining the rotation bound and the translation bound gives

$$\|\widehat{\mathbf{R}} - \mathbf{R}\|_F + \|\widehat{\mathbf{t}} - \mathbf{t}\|_2 \leq C(\mathcal{P}_s)\delta,$$

where $C(\mathcal{P}_s)$ depends on the scale and conditioning of the sampled point set. This proves Proposition 3.1.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly describe the main contributions of the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper describes the proposed architecture, training and inference procedures, data construction, evaluation protocol, benchmark settings, and key implementation details needed to reproduce the main results. Additional architecture and training details are provided in Appendix A and Appendix B; code and pretrained checkpoints will be released after acceptance.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The code has not been included in the initial submission because it is still being cleaned and documented for release. The datasets used in this paper are public benchmarks with documented access procedures, and we provide detailed architecture, training, inference, and evaluation details in the paper and appendix. Code and pretrained checkpoints will be released after acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 and Appendix B describe the datasets, perturbation settings, and evaluation metrics. Appendix A provides the architecture, input preprocessing, optimizer, learning-rate settings, batch size, and number of training epochs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experiment uses a fixed dataset and fixed seeds, and does not involve these.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper includes experiments and reports partial compute-resource information.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Yes, we have checked it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We provide the discussion of broader impacts in the appendix.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We used the publicly available dataset in our work and cited the corresponding papers in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: Yes, the training details of our model is presented.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our work does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our work does not involve human subjects research and therefore does not require IRB approval or equivalent review.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method of development in our work does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.