Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.081695

ARTICLE

Attention and Mamba Based Iterative Registration Network for Low-Overlap and Large-Scale Point Cloud

Haotian Cao[1,2] **and Qingsheng Zhu**[1,2,3,*]

[1]School of Astronomy and Space Science, University of Science and Technology of China, Hefei, China
[2]Nanjing Astronomical Instruments Research Center, Chinese Academy of Sciences, Nanjing, China
[3]CAS Nanjing Astronomical Instruments Co., Ltd., Nanjing, China

*Corresponding Author: Qingsheng Zhu. Email: qszhu@nairc.ac.cn

Received: 06 March 2026; Accepted: 27 April 2026; Published: 15 June 2026

ABSTRACT: Point Cloud Registration (PCR) is a basic task in computer vision, mobile robotics, and autonomous driving. PCR primarily faces challenges, including insufficient registration performance in low-overlap scenarios and high computational resource consumption in large-scale point cloud scenarios. Most recent PCR methods are transformer-based. Methods like transformers have quadratic computational complexity $\mathcal{O}(n^2d)$, leading to rapid increases in computational cost with large-scale point cloud data. To address these problems, an iterative PCR method named **A**ttention and **M**amba **B**ased **I**terative **R**egistration Network (**AMBIR**) is proposed, overcoming the shortcomings of the current PCR method on low-overlap and large-scale scenarios. Specifically, an iterative network architecture is introduced that learns overlap experience from prior registration results, thereby enhancing registration performance by leveraging knowledge from the preceding step. Additionally, to convert 3-D point cloud data into linear sequences suitable for the Mamba encoder, the Prior-Informed Co-aligned Serialization is proposed to ensure that points with adjacent indices after serialization are spatial neighbors, thereby improving the efficiency and robustness of the subsequent registration process. Lastly, a Consistency-Aware Mamba Encoder is introduced to leverage its linear computational complexity, making the method more suitable for large-scale point clouds. This method simultaneously overcomes the shortcomings of existing methods, including insufficient registration performance in low-overlap and large-scale point cloud scenarios. It performs well on the 3DMatch dataset, 3DLoMatch low-overlap dataset, and KITTI large-scale scene dataset, demonstrating high practical value.

KEYWORDS: Point cloud registration; deep learning; computer vision; attention mechanism; mamba model; iterative network

1 Introduction

Point cloud registration (PCR) is a basic task in computer vision, mobile robotics [1] and autonomous driving [2]. The task of PCR is to estimate an optimal transformation in the special Euclidean group $\mathfrak{se}(3)$ to align point clouds of the same object or environment captured from different coordinate systems into a unified coordinate frame [3]. Early PCR tasks primarily relied on geometric optimization. For example, the earliest geometric method is Iterative Closest Point (ICP) algorithm [4]. It employs an iterative optimization process that pairs points from two overlapping point clouds and minimizes pairwise Euclidean distances. ICP is easy to understand, but it requires precise initialization due to its non-convexity.

With the development of deep learning, especially the proposal of the attention mechanism [5], the Transformer model has been widely used in the PCR task. A very early transformer-based method is

CoFiNet [6], which proposes a coarse-to-fine registration network. It learns to match down-sampled nodes whose vicinity points share more overlap on a coarse scale, and it refines the corresponding relationships from the overlap area of the corresponding patches through an adaptive matching module at a fine scale. Further work, such as GeoTransformer [7], improves the ability of super-point discrimination by leveraging geometric relationships. CAST [8] designs a consistency-aware spot-guided Transformer, including a spot-guided cross-attention module to avoid interfering with irrelevant areas and a consistency-aware self-attention module to enhance matching capabilities with geometrically consistent correspondences. The transformer model can effectively model global spatial relationships and feature correspondences between point clouds, outperforming traditional methods based on geometric optimization and conventional CNNs.

While these Transformer or attention-based methods have shown distinctive performance, they still face two major shortcomings:

(1) In low-overlap scenarios, the low-overlap area makes it easier to extract features that appear similar on the surface but actually belong to different areas during the registration process, leading to a decline in registration performance;
(2) Transformer-based methods inherently have quadratic computational complexity $\mathcal{O}(n^2 d)$ with respect to sequence length, meaning computational costs increase dramatically as the sequence length grows, imposing strict computational demands and making them unfriendly to scenarios involving large-scale point clouds.

Recent works try to address shortcoming (1) using an iterative method. PEAL [9] introduces a post-processing method to refine the registration results, establishing an iterative approach by employing the same network as [7] repeatedly to demonstrate enhanced performance in low-overlap scenarios. AMR [10] considers the fact that the priors become increasingly accurate throughout the refinement steps, and proposes an iterative refinement network to leverage the knowledge of the overlap area, tailored for the low-overlap challenge in PCR. However, the iterative method requires multiple training iterations, which severely increases computational overhead.

State space models (SSMs) [11], especially the Mamba model [12], demonstrate extraordinary performance at efficiently capturing long-range contextual dependencies in sequence modeling tasks. Mamba leverages a linear-complexity state-space model to approximate global context, enhancing efficiency and scalability for long sequences and enabling it to address shortcoming (2). Its global receptive field and linear runtime enable fast, low-cost inference, ideal for large-scale or real-time applications. Unfortunately, Mamba is designed to handle sequential data, thereby leveraging its advantage of linear complexity. As 3-D data, point clouds exhibit spatial disorder and irregularity, so they require serialization before processing with Mamba.

To solve shortcomings above concurrently and enable the PCR network to be applicable to both low-overlap and large-scale point cloud scenarios, inspired by CAST [8], AMR [10] and Mamba [12], **A**ttention and **M**amba **B**ased **I**terative **R**egistration Network (**AMBIR**) is proposed, leveraging the attention mechanisms to suppress interference from irrelevant regions, the iterative model to extract features in low-overlap scenarios and the Mamba model to linearize its computational complexity simultaneously.

The effectiveness of this work stems from the following contributions:

- An iterative model that progressively learns overlap knowledge from prior registration to ground-truth alignment is incorporated, overcoming the performance degradation in low-overlap registration scenarios.
- A serialization method that converts 3-D point clouds into linear sequences is proposed to apply unordered and spatially irregular 3-D point cloud data to Mamba. While achieving linear computational complexity, serialization ensures that points at the same position in the sequence correspond spatially

by leveraging prior information to pre-align and uniformly sort the two point clouds, thereby improving the subsequent registration performance.

- A Mamba model with an overlap-driven soft gating mechanism and a bidirectional architecture is introduced. This model mitigates the computational resource consumption of partial attention mechanisms by achieving linear complexity. While achieving efficient global long-range feature aggregation, it endows the model with strong robustness to noise in non-overlapping regions through an implicit filtering mechanism.

2 Related Work

2.1 Transformer-Based PCR Method

Transformer-based PCR methods leverage the strong data-driven ability of the Transformer architecture for PCR. Numerous studies have incorporated encoder-decoder frameworks and attention mechanisms, significantly improving registration accuracy. In addition to the methods discussed in Section 1, OIF-Net [13] proposed a singular-intrinsic-point-based positional encoding approach for PCR networks. It employed a differentiable optimal transport layer to establish correspondences, which were then used to normalize each point for positional encoding, effectively eliminating issues arising from differing reference frames between the two point clouds. Additionally, it mitigated feature ambiguity and related problems by learning spatial consistency. RoITr [14] proposes a local-level attention mechanism embedded with point-pair feature coordinates to describe pose-invariant geometric structures. Based on this, it constructs a novel attention-based encoder-decoder architecture. At the global level, it introduces a global Transformer that learns rotation-invariant cross-frame spatial perception via a self-attention mechanism. This significantly enhances the feature discriminability and improves the model's robustness in low-overlap scenarios. SIRA-PCR [15] proposes the first method to explore simulation-to-reality adaptation in PCR. The framework incorporated an adaptive resampling module to address the domain gap between simulated and real point cloud patterns and constructed a synthetic scene-level PCR dataset that employed both physics-based and randomized strategies to arrange diverse objects. RegFormer [16] introduces a feature extraction Transformer and a bijective association Transformer, which capture long-range dependencies and filter outliers via global point feature extraction. This ensures high efficiency even in large-scale scenes while enabling the regression of initial transformations.

The method reviewed above leverages the advantages of Transformer models from different perspectives, improving the efficiency and speed of point cloud registration. However, none of these works overcome the inherent quadratic computational complexity $\mathcal{O}(n^2 d)$ of the model, still requiring significant computational performance during training and inference. This poses a challenge for running PCR networks on low-resource devices.

2.2 Iterative-Based Model

An iterative-based model gradually optimizes results, approaches targets, or solves problems by repeatedly executing fixed steps, using the output of the previous iteration as input for the next iteration. The well-known methods for the PCR task based on iterative models are PEAL [9] and AMR [10]. In addition, IFNet [17] proposes a novel iterative feedback network for unsupervised PCR, in which the representation of low-level features is efficiently enriched by rerouting subsequent high-level features. Besides the PCR task, iterative models have also been applied in many areas. In 3-D reconstruction, MSDER-MVS [18] optimizes depth estimation iteratively using residuals and the Jacobian without additional parameters. In point cloud completion, PMP-Net [19] achieves iterative refinement through shape deformation and builds point-level correspondences.

A drawback of the iterative-based model is that it introduces additional computational overhead due to the multiple iterative processes. Let the resource consumption of a single iteration be a, and the number of iterations be N. Then, the total resource consumption is aN. Particularly in complex scenarios such as large-scale, high-precision point clouds, where a can be very large, these models may consume excessive resources, limiting their use on low-resource devices.

2.3 Mamba-Based PCR Method

To apply the Mamba model to PCR, two issues need to be addressed: (1) how to convert 3-D point clouds into 1-D sequences; (2) how to extract global features. Recently, to apply Mamba to PCR, MT-PCR [20] performs Z-order-based spatial serialization on 3-D point cloud data, replaces the self-attention module in the CAST [8] backbone with a Mamba encoder, and constructs a hierarchical framework. This framework combines Mamba's global modeling capability with local attention and cross-scale optimization, reducing the VRAM usage of the registration network. E2MNet [21] replaces the feature extraction Transformer and bijection association Transformer in the RegFormer [16] backbone with the feature extraction Mamba2 module and spatio-temporal fusion module, respectively. It comprehensively captures the local and global features of point clouds and efficiently accomplishes large-scale PCR tasks. MaGo-I2P [22] proposes the first Mamba-based image-to-picture registration framework. It recovers the geometric structure of images through depth estimation, thereby constructing an implicit 3-D representation of the image scene to alleviate the modality gap between images and point clouds and facilitates cross-modal feature extraction. In specialized domains, AeroMamba [23] leveraged the Mamba architecture and a Hilbert curve to address the challenges posed by large-scale, featureless point clouds in aircraft assembly.

However, the aforementioned Mamba-based PCR methods are either optimized only for specific domains or designed only to address the registration of either large-scale or featureless point clouds. There is still no unified PCR method that can simultaneously address challenges in both large-scale and low-overlap scenarios. In addition, unlike permutation-invariant Transformers, the autoregressive nature of SSMs makes Mamba highly sensitive to sequence ordering. Consequently, rather than being a universal replacement, the $\mathcal{O}(n)$ efficacy of the Mamba model in 3-D vision is strictly conditioned on the ability of the pipeline to transform unordered spatial points into geometrically aligned linear sequences. This implies that a pure Mamba-based PCR method would be highly sensitive to sequence ordering and lack permutation invariance when processing unordered point clouds. Therefore, an attention and Mamba-based PCR network is significantly more advantageous.

3 Method

3.1 Problem Definition

The task of PCR is to transform point clouds of the same object or environment acquired under different coordinate systems into a single coordinate system by estimating an optimal special Euclidean group $\mathfrak{se}(3)$ [3]. Formally, the source point cloud is denoted as $\mathcal{P} = \{\boldsymbol{p}_i \in \mathbb{R}^3 \mid i = 1, \cdots, N\}$ and the target point cloud is denoted as $\mathcal{Q} = \{\boldsymbol{q}_j \in \mathbb{R}^3 \mid j = 1, \cdots, M\}$. Each point correspondence satisfies $(\boldsymbol{p}_i, \boldsymbol{q}_i) \in \mathcal{C}$ and has a weight w_i. PCR is to find an appropriate rotation matrix $\boldsymbol{R} \in \mathfrak{so}(3)$ and a translation vector $\boldsymbol{t} \in \mathbb{R}^3$, such that the following equation achieves a minimum value:

$$\min_{\boldsymbol{R},\boldsymbol{t}} \sum_{(\boldsymbol{p}_k,\boldsymbol{q}_k)\in\mathcal{C}} w_k \left\| \boldsymbol{R}\boldsymbol{p}_k + \boldsymbol{t} - \boldsymbol{q}_k \right\|_2^2 \tag{1}$$

3.2 Preliminaries

3.2.1 Attention Mechanism

Attention Mechanism [5] originated from research on the human visual system. When observing things, humans do not focus equally on all information; instead, they selectively concentrate on interesting or important parts, quickly capturing key information while ignoring irrelevant details. This mechanism was introduced into deep learning to improve the efficiency and accuracy of models when processing complex data. The attention mechanism includes self-attention, cross-attention, and multi-head attention.

Transformers are constructed by stacking self-attention and cross-attention modules, enabling effective modeling of global dependencies and feature correspondences. The self-attention mechanism computes attention weights over the same set of points to capture internal feature interactions, whereas cross-attention identifies correspondences between two different point sets. Formally, given a query $\boldsymbol{Q}$, key $\boldsymbol{K}$, and value $\boldsymbol{V}$, the attention can be computed as:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{2}$$

where d_k is the dimensionality of $\boldsymbol{K}$. According to Eq. (2), the computational complexity of the Transformer architecture is quadratic, i.e., $\mathcal{O}(n^2 d_k)$.

3.2.2 Iterative Model

Most existing PCR networks are based on a single training session. In contrast, iterative models undergo multiple training rounds. Specifically, a training count k is preset; during each training round, the model learns from the overlapping regions of the previous pre-existing model, and this process is repeated k times to obtain the final model.

The currently most effective iterative model is AMR [10]. Formally, taking AMR as an example, let $\mathcal{X} = \{\boldsymbol{R}, \boldsymbol{t}\}$, meaning that $\mathcal{X}$ is a combination of a rigid transformation. The iterative registration framework $\mathcal{F} = \{f_k(\mathcal{X}, \cdot) \mid k = 1, \cdots, K\}$ updates the model as:

$$\mathcal{X}_k = f_k(\mathcal{X}_{k-1}, \mathcal{P}, \mathcal{Q}) \tag{3}$$

which means that each model learns from the overlapping prior knowledge contained in the rigid transformation of the previous model. Eq. (3) follows the adaptive refinement paradigm established in [10], treating registration as a progressive residual update $\mathcal{X}_k = \Delta\mathcal{X}_k \circ \mathcal{X}_{k-1}$. By estimating a residual $\Delta\mathcal{X}_k$ that naturally decays toward the identity matrix $\boldsymbol{I}$ as alignment improves, the iterative process functions as a contraction mapping. This property theoretically ensures monotonic convergence and prevents the accumulation of registration errors across stages.

For iterative models, the registration accuracy varies at each step. Therefore, to train models for different steps separately, a transition function $\mathcal{T} = \{\epsilon\ (\mathcal{X}, \tau)\}$ is introduced into the model, where $\tau \in \{1, \cdots, T\}$ represents the accuracy level: the higher the value, the greater the accuracy, and when $\tau = T$, it corresponds to the ground-truth. For each step index k, $\epsilon\ (\mathcal{X}, k)$ produces a rigid transformation $\widetilde{\mathcal{X}}_{k-1}$ and further trains the registration network. Finally, the iterative model yields the final estimate $\hat{\mathcal{X}}$. The transition function $\mathcal{T} = \{\epsilon\ (\mathcal{X}, \tau)\}$ divides the transition state from the prior to the ground truth into $[1, \cdots, T]$, where each time step $\tau \in [1, \cdots, T]$ corresponds to a rotation matrix $\boldsymbol{R}_\tau$ and a translation vector $\boldsymbol{t}_\tau$.

Since rotation matrices are nonlinear, the spherical interpolation function is used:

$$q_\tau = \mathrm{Slerp}(q_{\mathrm{prior}}, q_{\mathrm{gt}}; \alpha_\tau) = \frac{\sin((1-\alpha_\tau)\Omega)}{\sin(\Omega)} q_{\mathrm{prior}} + \frac{\sin(\alpha_\tau \Omega)}{\sin(\Omega)} q_{\mathrm{gt}} \tag{4}$$

where q_τ, q_{prior} and q_{gt} are quaternions of $\boldsymbol{R}_\tau$, $\boldsymbol{R}_{\mathrm{prior}}$ and $\boldsymbol{R}_{\mathrm{gt}}$, $\Omega = \arccos(q_{\mathrm{prior}} \cdot q_{\mathrm{gt}})$ and $\alpha_\tau = \frac{\tau}{T}$. Then, convert q_τ into $\boldsymbol{R}_\tau$.

Linear interpolation is employed for the translation vectors:

$$\boldsymbol{t}_\tau = (1-\alpha_\tau) \cdot \boldsymbol{t}_{\mathrm{prior}} + \alpha_\tau \cdot \boldsymbol{t}_{\mathrm{gt}} \tag{5}$$

3.2.3 SSMs and Mamba

SSM consists of two equations: the state equation and the observation equation:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{x}(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{x}(t) \end{cases} \tag{6}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^L$ is system input, L is input dimension and $\boldsymbol{y}(t) \in \mathbb{R}^N$ is hidden state, N is state dimension, $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is state matrix, $\boldsymbol{B} \in \mathbb{R}^{N \times L}$ is input matrix, $\boldsymbol{C} \in \mathbb{R}^{L \times N}$ is output matrix and $\boldsymbol{D} \in \mathbb{R}^{L \times L}$ is feed-through matrix.

Computers are adept at processing discrete signals, and in modern control theory, the zero-order hold (ZOH) is used to convert them into discrete-time state space models. Denote the sampling period as Δt, then the state matrix and the hidden matrix are converted to:

$$\begin{cases} \overline{\boldsymbol{A}} = \mathrm{e}^{\boldsymbol{A}\Delta t} \\ \overline{\boldsymbol{B}} = \displaystyle\int_0^{\Delta t} \mathrm{e}^{\boldsymbol{A}\tau} \mathrm{d}\tau \cdot \boldsymbol{B} = \left(\mathrm{e}^{\boldsymbol{A}\Delta t} - \boldsymbol{I}\right)(\boldsymbol{A}\Delta t)^{-1} \Delta t \cdot \boldsymbol{B} \end{cases} \tag{7}$$

Therefore, the discrete-time state space model can be expressed as:

$$\begin{cases} \boldsymbol{x}_k = \overline{\boldsymbol{A}}\boldsymbol{x}_{k-1} + \overline{\boldsymbol{B}}\boldsymbol{x}_k \\ \boldsymbol{y}_k = \overline{\boldsymbol{C}}\boldsymbol{h}_k + \overline{\boldsymbol{D}}\boldsymbol{x}_k \end{cases} \tag{8}$$

where k is the discrete time step.

Mamba model, proposed in [12] and inspired by SSMs, extends them into selective state space models (Selective SSMs). In Mamba, the parameters $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, Δt, originally defined over the core dimensions of state size N and sequence length L, are augmented with the batch size B and model dimension D, resulting in the parameter set $\widetilde{\boldsymbol{A}}, \widetilde{\boldsymbol{B}}, \widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{T}}$ in Mamba, and these parameters are adaptively adjusted based on the input $\boldsymbol{x}_k$. To address the parallelization challenge of the single linear model, Mamba employs a global convolution expansion:

$$\begin{cases} \boldsymbol{K} = \left(\widetilde{\boldsymbol{C}}\overline{\widetilde{\boldsymbol{B}}}, \widetilde{\boldsymbol{C}}\overline{\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{B}}}, \ldots, \widetilde{\boldsymbol{C}}\overline{\widetilde{\boldsymbol{A}}}^{L-1}\overline{\widetilde{\boldsymbol{B}}}\right) \\ \boldsymbol{y} = \boldsymbol{x} * \boldsymbol{K} \end{cases} \tag{9}$$

where $\boldsymbol{K}$ is global convolution kernel. In this way, Mamba achieves a linear complexity, addressing the computational bottleneck of the quadratic complexity from the Transformer architecture.

3.3 Network Architecture

Mamba can leverage its linear computational complexity to reduce the computational burden caused by the Transformer architecture, the serialization method can convert 3-D point clouds into linear sequences and apply them to the Mamba model and the iterative-based model can continuously learn knowledge from overlapping priors and ground truth. By combining the strengths of these three models, it can effectively reduce the computational cost of point cloud registration while addressing large-scale and low-overlap point cloud alignment.

To combine these three models, AMBIR has two parts: **iteration backbone** and **iteration process**. The iteration backbone refers to the method used at each step of the overall network. After this iteration is completed, the process proceeds to the next iteration in accordance with the iteration process. Therefore, for the entire registration network to operate efficiently, these two components need to work in tandem. For the iteration backbone, it must achieve high registration performance while maintaining low resource usage. This ensures that the model's performance improves after multiple iterations without excessive resource consumption. For the iteration process, appropriate learning rules need to be designed so that the model can learn the registration knowledge from the previous iteration at each step.

3.3.1 Iteration Backbone

As shown in Fig. 1, the iteration backbone of AMBIR consists of feature extraction, hybrid coarse registration, and sparse-to-dense fine registration in sequence.

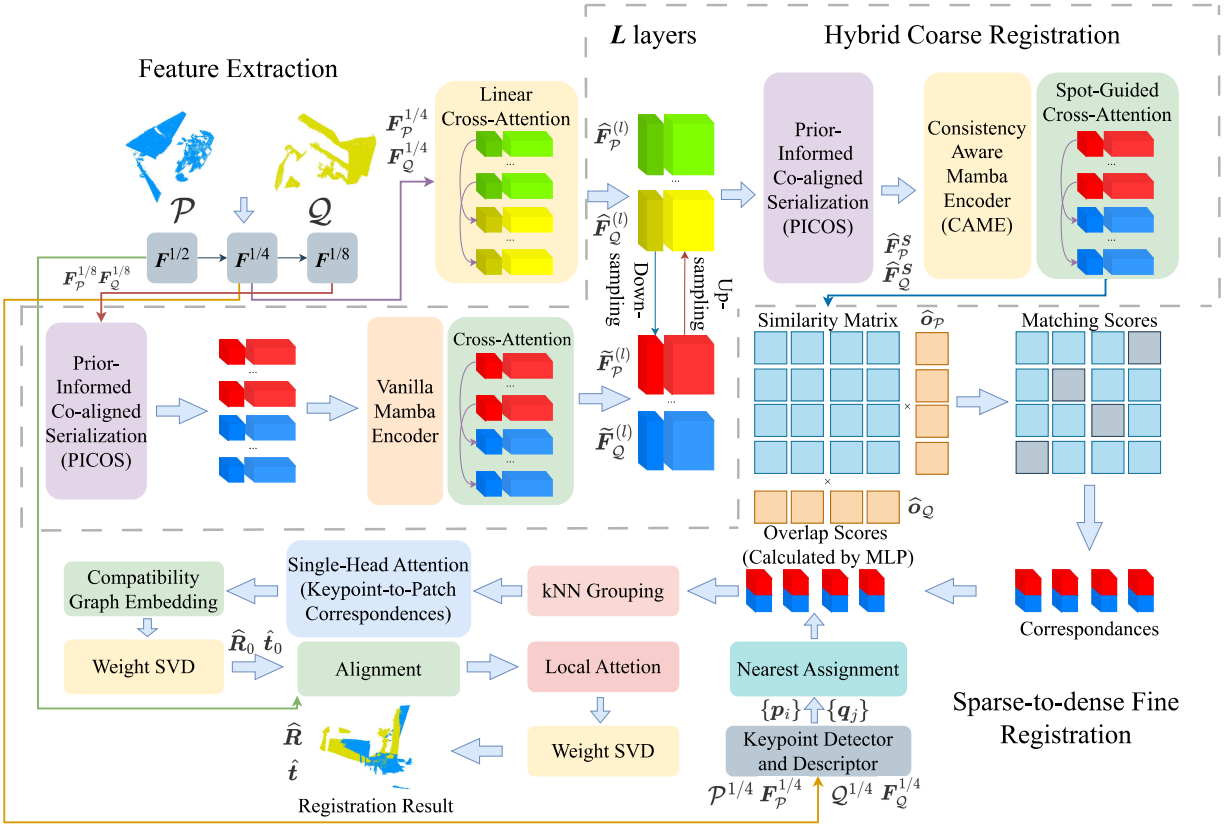

Figure 1: Overview of the iterative backbone of AMBIR.

Feature Extraction. Feature extraction is the process of deriving low-dimensional representations with discriminability, invariance, and compactness from raw point clouds. Herein, FA-KPConv [24] is utilized to encode the input point clouds into multi-scale feature representations. Let the feature map of the original point clouds be denoted as $\boldsymbol{F} = \{\boldsymbol{F}_{\mathcal{P}}, \boldsymbol{F}_{\mathcal{Q}}\}$, where $\boldsymbol{F}_{\mathcal{P}}$ and $\boldsymbol{F}_{\mathcal{Q}}$ are the feature maps of the original point clouds $\mathcal{P}$ and $\mathcal{Q}$, respectively. For sampling levels $k = 0, 1, 2, \cdots$, the corresponding decoded sampled feature map is $\boldsymbol{F}^{1/k} = \{\boldsymbol{F}_{\mathcal{P}}^{1/k}, \boldsymbol{F}_{\mathcal{Q}}^{1/k}\}$. In particular, $\boldsymbol{F}^{1/4}$ is called semi-dense feature and $\boldsymbol{F}^{1/8}$ is called coarse feature. The down-sampled key points at the topmost level are referred to as superpoints, which serve as anchor points for subsequent stages.

Hybrid Coarse Registration. First, to enhance the semi-dense feature, a linear cross-attention [25] is adopted before subsequent modules. Both semi-dense feature and coarse feature superpoints need to be serialized for subsequent processing. Thus, **Prior-Informed Co-aligned Serialization (PICOS)** is applied to bridge the gap between unordered point clouds and sequential models. Utilizing the transformation estimate $\mathcal{X}_{k-1}$ from the previous iteration as a structural prior, the source and target point clouds are projected into semantically aligned 1-D sequences via a shared Hilbert curve, ensuring that geometrically corresponding points are mapped to proximate indices. Then, coarse feature and semi-dense feature are processed differently in the network, but can be fused [26] to enhanced semi-dense feature further:

$$\begin{aligned}\hat{\boldsymbol{F}}^{1/4} &= \boldsymbol{F}^{1/4} + \mathrm{MLP}(\text{Nearest Up-sampling}(\boldsymbol{F}^{1/8})) \\ \hat{\boldsymbol{F}}^{1/8} &= \boldsymbol{F}^{1/8} + \mathrm{MLP}(\text{Interpolated Down-sampling}(\boldsymbol{F}^{1/4}))\end{aligned} \tag{10}$$

For coarse feature, these synchronized sequences are passed through an encoder composed of H stacked Mamba blocks to extract hierarchical geometric features. Each block consists of layer normalization (LN), a selective state space model (SelectiveSSM) [12], depth-wise separable convolutions (DW) [27], and residual connections. The architecture is illustrated in Fig. 2, and the h-th block of H Mamba blocks can be described as:

$$\begin{aligned}\boldsymbol{F}'_{h-1} &= \mathrm{LN}(\boldsymbol{F}_{h-1}) \\ \boldsymbol{F}'_{h} &= \sigma\left(\mathrm{DW}\left(\mathrm{Linear}\left(\boldsymbol{F}'_{h-1}\right)\right)\right) \\ \boldsymbol{F}''_{h} &= \sigma\left(\mathrm{Linear}\left(\boldsymbol{F}'_{h-1}\right)\right) \\ \boldsymbol{F}_{h} &= \mathrm{Linear}\left(\mathrm{SelectiveSSM}(\boldsymbol{F}'_{h}) \odot \boldsymbol{F}''_{h}\right) + \boldsymbol{F}_{h-1}\end{aligned} \tag{11}$$

where σ represents the SiLU activation function [28].

For semi-dense features, sequences are fed into the **Consistency-Aware Mamba Encoder (CAME)**. As a lightweight alternative to the computationally expensive self-attention mechanism, CAME employs a bi-directional SSM to aggregate global geometric context with linear complexity. Crucially, an overlap-driven soft-gating mechanism is integrated to implicitly suppress features from non-overlapping regions, enhancing robustness against outliers. Finally, the enhanced semi-dense features $\hat{\boldsymbol{F}}_{\mathcal{P}} \in \mathbb{R}^{M' \times D}$ and $\hat{\boldsymbol{F}}_{\mathcal{Q}} \in \mathbb{R}^{N' \times D}$ are reverted to their original spatial order and forwarded to the Spot-Guided Cross-Attention module inherited from CAST [8]. This module performs explicit feature interaction by restricting attention computations to local consistent regions, aka spots, thereby generating a reliable similarity matrix $\boldsymbol{S} = \hat{\boldsymbol{F}}_{\mathcal{P}} \hat{\boldsymbol{F}}_{\mathcal{Q}}^{\mathrm{T}}$. Furthermore, $\hat{\boldsymbol{F}}_{\mathcal{P}}$ and $\hat{\boldsymbol{F}}_{\mathcal{Q}}$ are fed into a point-wise multilayer perceptron to calculate overlap scores by:

$$P_{ij} = \hat{o}_i^{\mathcal{P}} \hat{o}_j^{\mathcal{Q}} \cdot \mathrm{softmax}_{k \in \{1, \cdots, M'\}}(\boldsymbol{S}_{kj})_i \cdot \mathrm{softmax}_{k \in \{1, \cdots, N'\}}(\boldsymbol{S}_{ik})_j \tag{12}$$

where $\hat{o}_i^{\mathcal{P}}$ and $\hat{o}_j^{\mathcal{Q}}$ are overlap scores of the i-th node of and the j-th node of semi-dense feature, respectively.

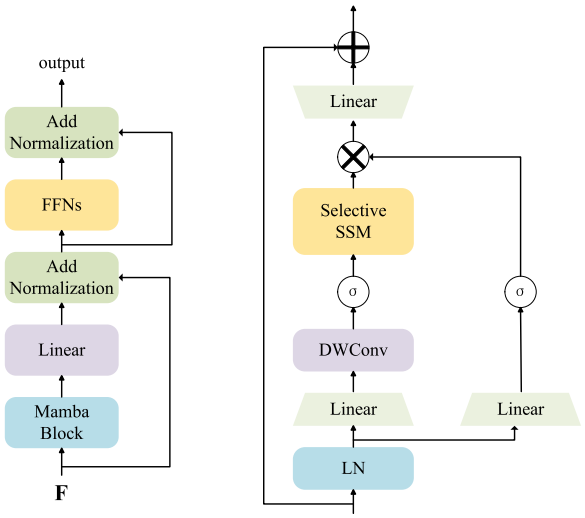

Figure 2: Architecture of the Mamba encoder and Mamba block. **Left**: Mamba Encoder with residual connections and feedforward neural networks (FNNs). **Right**: Mamba block centering around the SelectiveSSM.

Sparse-to-Dense Fine Registration. Inspired by the hierarchical strategy [8], the fine registration module employs a lightweight sparse-to-dense mechanism to achieve precise alignment without computational bottlenecks. Distinct keypoints are extracted from local patches centered at semi-dense nodes of $\boldsymbol{X}^{1/4}$, and their virtual correspondences $\hat{\boldsymbol{y}}_i$ are predicted by aggregating features from corresponding patches in $\boldsymbol{Y}^{1/4}$ via a single-head attention layer. To ensure geometric robustness, a compatibility graph constructs spatial consistency constraints, enabling graph embedding layers to predict confidence weights for these sparse matches and yield an initial alignment via the weighted Kabsch algorithm. Subsequently, dense point-to-point correspondences are established between $\boldsymbol{X}^{1/2}$ and $\boldsymbol{Y}^{1/2}$ through local attention within a dynamic radius relative to the initial alignment, allowing the final incremental transformation $\Delta\mathcal{X}_k$ to be solved analytically over the joint set of weighted sparse and dense correspondences.

Notably, although the sparse-to-dense fine registration introduces local attention and matching costs, it computes only within local neighborhoods or on dynamically down-sampled point sets. As a result, they effectively avoid the time complexity introduced by global attention.

3.3.2 Iteration Process

As shown in Fig. 3, the registration pipeline is structured as a cascade of K adaptive refinement stages, where each stage employs an identical network architecture based on the proposed backbone but possesses independent trainable parameters tailored to specific noise distributions. To foster adaptivity, synthetic prior transformations spanning T discrete accuracy levels are generated and linearly partitioned into K groups ($K < T$), with the specific model index k for a given accuracy level τ assigned via $k = \lceil (\tau/T) * K \rceil$, where $\lceil \cdot \rceil$ denotes the ceiling function. During the inference phase, iterative optimization begins with an initial rigid transformation $\mathcal{X}_0$, which can be derived from vanilla CAST [8] rather than a random initialization. The permutation-invariance of CAST enables it to provide a reliable initial transformation without serialization, effectively avoiding problems caused by poor initialization. Then, $\mathcal{X}_0$ proceeds recursively, where the transformation estimation $\mathcal{X}_k$ output by the current stage functions as the structural prior for the subsequent stage, culminating in the final high-precision alignment $\mathcal{X}_K$ produced by the last refinement model.

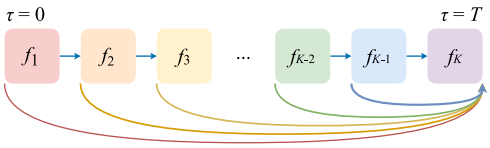

Figure 3: Iterative process of AMBIR.

3.4 Prior-Informed Co-Aligned Serialization

SSMs, particularly Mamba, rely on autoregressive modeling of 1-D sequences to capture global context with linear complexity. Bridging the dimensional gap between unordered 3-D point clouds and ordered 1-D sequences is a fundamental prerequisite for the proposed architecture. Thus, a serialization strategy that transforms the challenging global registration problem into a manageable local sequence matching task is needed.

Standard point cloud serialization typically employs Space-Filling Curves (SFCs), such as the Hilbert curve or Z-order curve, to map 3-D coordinates onto a 1-D manifold while preserving local neighborhood structures [20]. While effective for static tasks such as semantic segmentation, where the coordinate frame is fixed, SFCs exhibit a critical limitation in registration scenarios due to their rotational sensitivity. SFCs are strictly coordinate-dependent. A rigid transformation $\mathcal{T} \in \mathfrak{se}(3)$ applied to a point cloud induces a non-linear distortion on the traversal path of the SFC. Consequently, two points $\boldsymbol{p}_i \in \mathcal{P}$ and $\boldsymbol{q}_j \in \mathcal{Q}$ that correspond to the same physical geometry may be mapped to drastically distant indices in their respective generated sequences, i.e., $|\mathrm{idx}(\boldsymbol{p}_i) - \mathrm{idx}(\boldsymbol{q}_j)| \gg 0$ due to the relative pose between $\mathcal{P}$ and $\mathcal{Q}$. This phenomenon is termed Semantic Misalignment. In the absence of alignment, the Mamba encoder is forced to implicitly learn complex, long-range correlations to associate corresponding features, effectively wasting its selective scanning capacity on solving global rotational invariance rather than focusing on local feature refinement. To address this problem, PICOS is proposed, which consists of three components: proxy alignment, shared projection and siamese reordering. Fig. 4 illustrates the architecture of PICOS.

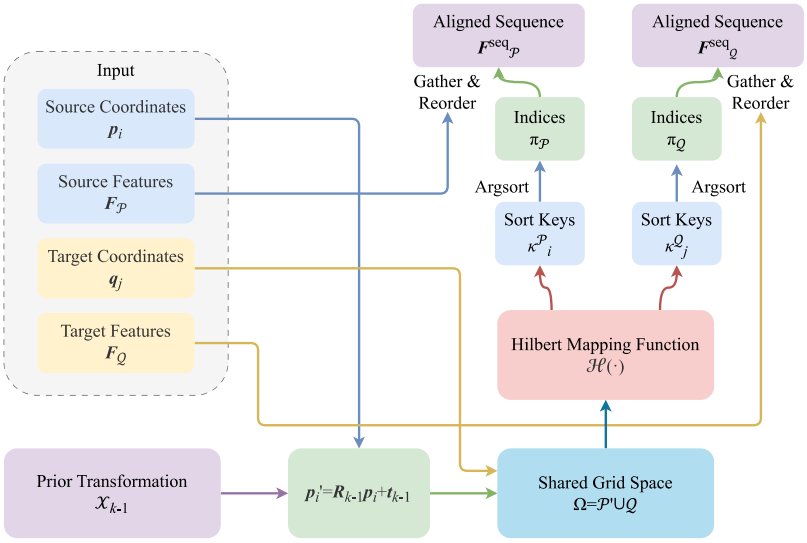

Figure 4: Architecture of PICOS.

3.4.1 Proxy Alignment and Shared Projection

To resolve this bottleneck, the strategy decouples the spatial ordering from the feature representation. The transformation estimate $\mathcal{X}_{k-1}$ from the previous iteration is utilized as a structural prior to synchronize the scanning order of the two point clouds.

Formally, the process begins by constructing a proxy point cloud $\mathcal{P}'$:

$$\boldsymbol{p}'_i = \boldsymbol{R}_{k-1}\boldsymbol{p}_i + \boldsymbol{t}_{k-1}, \quad \forall \boldsymbol{p}_i \in \mathcal{P} \tag{13}$$

It is crucial to note that $\mathcal{P}'$ is used solely to calculate the sorting keys. The actual input features $\boldsymbol{F}_{\mathcal{P}}$ remain associated with the original local coordinates to prevent label leakage and ensure that the network learns the residual update $\Delta\mathcal{X}_k$.

Next, a shared canonical space Ω is established, defined as a discretized 3-D grid bounding the union of $\mathcal{P}'$ and $\mathcal{Q}$. A Hilbert mapping function $\mathcal{H} : \mathbb{R}^3 \to \mathbb{Z}$ is defined to project 3-D coordinates to 1-D integers. The sorting keys for both clouds are computed within this unified frame:

$$\kappa_i^{\mathcal{P}} = \mathcal{H}(\boldsymbol{p}'_i), \quad \kappa_j^{\mathcal{Q}} = \mathcal{H}(\boldsymbol{q}_j) \tag{14}$$

Because $\mathcal{P}'$ is coarsely aligned with $\mathcal{Q}$ by the prior, spatially overlapping regions are guaranteed to occupy identical or adjacent cells in Ω, resulting in synchronized Hilbert keys.

3.4.2 Siamese Reordering

Finally, obtain the permutation indices $\pi_{\mathcal{P}}$ and $\pi_{\mathcal{Q}}$ by sorting the keys $\boldsymbol{\kappa}^{\mathcal{P}}$ and $\boldsymbol{\kappa}^{\mathcal{Q}}$ in ascending order:

$$\pi_{\mathcal{P}} = \text{argsort}(\{\kappa_i^{\mathcal{P}} \mid i = 1, \cdots, N\}), \quad \pi_{\mathcal{Q}} = \text{argsort}(\{\kappa_j^{\mathcal{Q}} \mid j = 1, \cdots, M\}) \tag{15}$$

It is worth noting that while the `argsort` operation introduces a theoretical time complexity of $\mathcal{O}(n \log n)$, sorting linear arrays is highly parallelizable and heavily optimized on modern GPUs. This serialization step acts as a lightweight geometric proxy, explicitly avoiding the $\mathcal{O}(n^2)$ bottleneck typically associated with dense compatibility graph construction.

The input features $\boldsymbol{F}_{\mathcal{P}}$ and $\boldsymbol{F}_{\mathcal{Q}}$ are reordered according to $\pi_{\mathcal{P}}$ and $\pi_{\mathcal{Q}}$ before being fed into the Mamba encoder. This operation is formally expressed as:

$$\begin{aligned} \boldsymbol{F}_{\mathcal{P}}^{\text{seq}} &= \text{Gather}(\boldsymbol{F}_{\mathcal{P}}, \pi_{\mathcal{P}}), \quad \boldsymbol{p}^{\text{seq}} = \text{Gather}(\boldsymbol{p}, \pi_{\mathcal{P}}) \\ \boldsymbol{F}_{\mathcal{Q}}^{\text{seq}} &= \text{Gather}(\boldsymbol{F}_{\mathcal{Q}}, \pi_{\mathcal{Q}}), \quad \boldsymbol{q}^{\text{seq}} = \text{Gather}(\boldsymbol{q}, \pi_{\mathcal{Q}}) \end{aligned} \tag{16}$$

where $\text{Gather}(\boldsymbol{F}, \pi)$ means rearranging the rows, also features in $\boldsymbol{F}$ according to the order specified by the permutation index π.

Eq. (16) creates a siamese sequence pair where the n-th token in the source sequence and the n-th token in the target sequence likely represent geometrically neighboring regions.

In summary, PICOS enables the full selective scanning capability of the Mamba architecture to focus on comparing local features rather than learning global rotation invariance in $\mathfrak{so}(3)$, thereby significantly improving feature-extraction accuracy. This is a property not possessed by Hilbert and Z-order curves. The pre-alignment mechanism can leverage GPU parallelism to achieve better registration performance with only a modest increase in computational overhead. In addition, in low-overlap scenarios, the shared

projection ensures that the non-overlapping segments are clustered together in the sequence. This clustering enables the subsequent soft-gating mechanism to efficiently identify and suppress these segments as contiguous noise blocks, rather than scattered outliers.

3.5 Consistency-Aware Mamba Encoder

While the serialization (Section 3.4) provides a geometrically aligned token sequence, standard SSMs treat all tokens equally during the recurrent state update. In the context of registration, however, points in non-overlapping yet similar-but-distinct regions (e.g., flat walls and desktops) act as noise, potentially contaminating the global context. The original CAST [8] addressed this by using a graph-based sampling strategy that applied sparse self-attention only to consistent nodes. Although effective, constructing compatibility graphs incurs $O(n^2)$ computational complexity or requires complex indexing overhead. To circumvent this bottleneck while retaining the ability to filter outliers, the Consistency-Aware Mamba Encoder (CAME) is proposed. As illustrated in Fig. 5, CAME leverages the selective scanning capability of Mamba to implicitly gate the information flow based on overlap confidence, achieving robust feature aggregation with linear complexity. The encoder consists of three parts: coordinate-injected embedding, consistency-guided soft gating, and bi-directional aggregation:

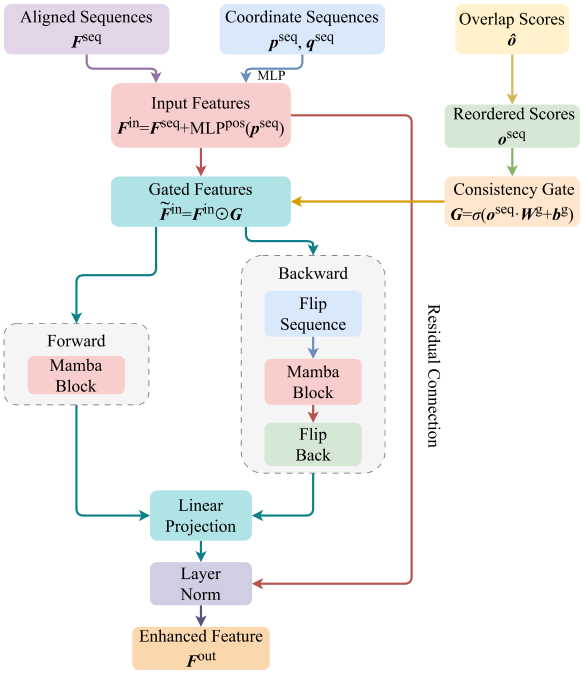

Figure 5: Architecture of CAME. Note: Some relevant mathematical symbols representing both the source point cloud $\mathcal{P}$ and the target point cloud $\mathcal{Q}$ have been simplified from two to one, e.g., $\boldsymbol{F}_{\mathcal{P}}^{\text{seq}}$, $\boldsymbol{F}_{\mathcal{Q}}^{\text{seq}}$ are simplified to $\boldsymbol{F}^{\text{seq}}$.

3.5.1 Coordinate-Injected Embedding

Although the input features are ordered via the Hilbert curve, the standard Mamba architecture processes sequences based strictly on relative positions within the 1-D array, lacking explicit awareness of the underlying 3-D metric space. To compensate for this loss of metric information during serialization, geometric embedding is injected prior to feature aggregation.

Let $\boldsymbol{F}_{\mathcal{P}}^{\text{seq}}, \boldsymbol{F}_{\mathcal{Q}}^{\text{seq}} \in \mathbb{R}^{N\times C}$ denote the feature sequence ordered by the permutation π, and $\boldsymbol{p}^{\text{seq}}, \boldsymbol{q}^{\text{seq}} \in \mathbb{R}^{N\times 3}$ denote the corresponding coordinate sequence. A position-encoding multi-layer perceptron MLP^{pos} maps

the physical coordinates into the high-dimensional feature space. This embedding is added to the semantic features via a residual connection:

$$\begin{aligned} \boldsymbol{F}_{\mathcal{P}}^{\text{in}} &= \boldsymbol{F}_{\mathcal{P}}^{\text{seq}} + \text{MLP}^{\text{pos}}(\boldsymbol{p}^{\text{seq}}) \\ \boldsymbol{F}_{\mathcal{Q}}^{\text{in}} &= \boldsymbol{F}_{\mathcal{Q}}^{\text{seq}} + \text{MLP}^{\text{pos}}(\boldsymbol{q}^{\text{seq}}) \end{aligned} \tag{17}$$

This operation ensures that the SSM implicit states can leverage both the sequential context and the absolute spatial distribution, facilitating the learning of distance-dependent geometric dependencies.

3.5.2 Consistency-Guided Soft Gating

To emulate the outlier-rejection capability of graph-based sampling without incurring quadratic computational costs, a soft gating mechanism is introduced. This mechanism dynamically modulates the information flow into the SSM based on the estimated reliability of each point.

Let $\hat{\boldsymbol{o}}_{\mathcal{P}}, \hat{\boldsymbol{o}}_{\mathcal{Q}} \in \mathbb{R}^N$ represent the point-wise overlap scores predicted by the segmentation head. These scores are first reordered according to the serialization index π to obtain $\boldsymbol{o}_{\mathcal{P}}^{\text{seq}}, \boldsymbol{o}_{\mathcal{Q}}^{\text{seq}}$. A learnable consistency gate $\boldsymbol{G} \in \mathbb{R}^{N \times C}$ is then generated via a linear projection followed by a Sigmoid activation:

$$\begin{aligned} \boldsymbol{G}_{\mathcal{P}} &= \sigma(\boldsymbol{o}_{\mathcal{P}}^{\text{seq}} \cdot \boldsymbol{W}_{\mathcal{P}}^{\text{g}} + \boldsymbol{b}_{\mathcal{P}}^{\text{g}}) \\ \boldsymbol{G}_{\mathcal{Q}} &= \sigma(\boldsymbol{o}_{\mathcal{Q}}^{\text{seq}} \cdot \boldsymbol{W}_{\mathcal{Q}}^{\text{g}} + \boldsymbol{b}_{\mathcal{Q}}^{\text{g}}) \end{aligned} \tag{18}$$

where $\boldsymbol{W}_{\mathcal{P}}^{\text{g}}, \boldsymbol{W}_{\mathcal{Q}}^{\text{g}}$ and $\boldsymbol{b}_{\mathcal{P}}^{\text{g}}, \boldsymbol{b}_{\mathcal{Q}}^{\text{g}}$ are trainable parameters. The input features are modulated via element-wise multiplication:

$$\begin{aligned} \widetilde{\boldsymbol{F}}_{\mathcal{P}}^{\text{in}} &= \boldsymbol{F}_{\mathcal{P}}^{\text{in}} \odot \boldsymbol{G}_{\mathcal{P}} \\ \widetilde{\boldsymbol{F}}_{\mathcal{Q}}^{\text{in}} &= \boldsymbol{F}_{\mathcal{Q}}^{\text{in}} \odot \boldsymbol{G}_{\mathcal{Q}} \end{aligned} \tag{19}$$

For theoretical insight, the core recurrence of an SSM is governed by the state equation $\boldsymbol{x}_k = \overline{\boldsymbol{A}}\boldsymbol{x}_{k-1} + \overline{\boldsymbol{B}}\boldsymbol{x}_k$. By modulating the input $\boldsymbol{x}_k$ with the gate $\boldsymbol{G}$, here $\boldsymbol{x}_k$ is $\widetilde{\boldsymbol{F}}_{\text{in}}$, the magnitude of feature vectors corresponding to non-overlapping regions, where $\boldsymbol{G} \approx 0$ is suppressed. Consequently, the hidden state $\boldsymbol{x}_k$ is updated primarily by features from consistent, high-overlap regions, effectively preventing background noise from propagating into the long-range global context.

3.5.3 Bi-Directional Aggregation

Space-filling curves impose a fixed traversal direction, which introduces a directional bias in information propagation. To capture the full geometric context and ensure isotropic feature learning, a Bi-directional Mamba strategy is employed. The gated sequences $\widetilde{\boldsymbol{F}}_{\mathcal{P}}^{\text{in}}, \widetilde{\boldsymbol{F}}_{\mathcal{Q}}^{\text{in}}$ are each processed by two independent SSM blocks scanning in opposite directions:

$$\begin{aligned} \boldsymbol{Y}_{\mathcal{P}}^{\text{fwd}} &= \text{SSM}^{\text{fwd}}(\widetilde{\boldsymbol{F}}_{\mathcal{P}}^{\text{in}}), & \boldsymbol{Y}_{\mathcal{Q}}^{\text{fwd}} &= \text{SSM}^{\text{fwd}}(\widetilde{\boldsymbol{F}}_{\mathcal{Q}}^{\text{in}}) \\ \boldsymbol{Y}_{\mathcal{P}}^{\text{bwd}} &= \text{flip}(\text{SSM}^{\text{bwd}}(\text{flip}(\widetilde{\boldsymbol{F}}_{\mathcal{P}}^{\text{in}}))), & \boldsymbol{Y}_{\mathcal{Q}}^{\text{bwd}} &= \text{flip}(\text{SSM}^{\text{bwd}}(\text{flip}(\widetilde{\boldsymbol{F}}_{\mathcal{Q}}^{\text{in}}))) \end{aligned} \tag{20}$$

here, $\text{flip}(\cdot)$ denotes the operation of reversing the sequence order. The forward scan captures dependencies from the start of the Hilbert curve, while the backward scan aggregates context from the end, ensuring that every point receives global information.

3.5.4 Feature Fusion and Restoration

The context-enriched features from both directions are fused via element-wise addition. To facilitate gradient flow and preserve original semantic information, a residual connection with the pre-gated input is applied, followed by normalization:

$$\begin{aligned} \boldsymbol{F}_{\mathcal{P}}^{\text{out}} &= \text{LayerNorm}(\boldsymbol{F}_{\mathcal{P}}^{\text{in}} + \text{Linear}(\boldsymbol{\Upsilon}_{\mathcal{P}}^{\text{fwd}} + \boldsymbol{\Upsilon}_{\mathcal{P}}^{\text{bwd}})) \\ \boldsymbol{F}_{\mathcal{Q}}^{\text{out}} &= \text{LayerNorm}(\boldsymbol{F}_{\mathcal{Q}}^{\text{in}} + \text{Linear}(\boldsymbol{\Upsilon}_{\mathcal{Q}}^{\text{fwd}} + \boldsymbol{\Upsilon}_{\mathcal{Q}}^{\text{bwd}})) \end{aligned} \tag{21}$$

Finally, to maintain compatibility with downstream modules that rely on the original point cloud indexing (such as the explicit cross-attention module), the output sequences $\boldsymbol{F}_{\mathcal{P}}^{\text{out}}, \boldsymbol{F}_{\mathcal{Q}}^{\text{out}}$ are mapped back to the original spatial order using the inverse permutation π^{-1}. This restoration step renders the CAME module structurally transparent to the rest of the network, functioning as a highly efficient, plug-and-play module that achieves comparable global routing to sparse self-attention without the quadratic cost.

In summary, the overall effective theoretical complexity of AMBIR across multiple iterations can be rigorously expressed as $\mathcal{O}(K \cdot (n \log n + \text{local attention} + \text{matching}))$. In existing iterative frameworks, global feature aggregation typically introduces a complexity of $\mathcal{O}(K \cdot n^2)$. By contrast, the $\mathcal{O}(n)$ complexity of the Mamba backbone mitigates this issue, allowing AMBIR to maintain low time complexity even over multiple iterative rounds.

3.6 Loss Functions

To supervise the iterative refinement framework, a multi-task objective function is designed, structured into four components: keypoint detection $\mathcal{L}_{\text{kd}}$, coarse matching $\mathcal{L}_{\text{cm}}$, keypoint matching $\mathcal{L}_{\text{km}}$, and dense registration $\mathcal{L}_{\text{dr}}$. The total loss $\mathcal{L}$ is a weighted sum of the stage-wise losses over K iterations:

$$\mathcal{L} = \sum_{k=1}^{K} \gamma_k \left(\mathcal{L}_{\text{kd}} + \mathcal{L}_{\text{cm}}^{(k)} + \mathcal{L}_{\text{km}}^{(k)} + \mathcal{L}_{\text{dr}}^{(k)} \right) \tag{22}$$

where γ_k denotes the stage-specific weight.

3.6.1 Keypoint Detection

Inspired by Usip [29], define the loss function of keypoint detection as:

$$\mathcal{L}_{\text{kd}} = \frac{1}{N} \sum_{i=1}^{N} \left(\log \widetilde{\sigma}_i + \frac{|\boldsymbol{p}_i - \boldsymbol{q}_{j^*(i)}|}{\widetilde{\sigma}_i} \right) + \frac{1}{M} \sum_{j=1}^{M} \left(\log \widetilde{\sigma}_j + \frac{|\boldsymbol{q}_j - \boldsymbol{p}_{i^*(j)}|}{\widetilde{\sigma}_j} \right) \tag{23}$$

where $\boldsymbol{p}_i \in \mathbb{R}^3$ and $\boldsymbol{q}_j \in \mathbb{R}^3$ are the key points from the source point cloud $\mathcal{P}$ and the target point cloud $\mathcal{Q}$, respectively. The indices $j^*(i) = \arg\min_j \|\boldsymbol{p}_i - \boldsymbol{q}_j\|$ and $i^*(j) = \arg\min_i \|\boldsymbol{q}_j - \boldsymbol{p}_i\|$ denote the nearest neighbors. The weight term $\widetilde{\sigma}$ for perceptual uncertainty is calculated as the average predicted variance of two matched key points, i.e., $\widetilde{\sigma}_i = \frac{1}{2}\left(\sigma_{p,i} + \sigma_{q,j^*(i)}\right)$ and $\widetilde{\sigma}_j = \frac{1}{2}\left(\sigma_{q,j} + \sigma_{p,i^*(j)}\right)$. This loss aligns the predicted key points of the source and target point clouds in space while accounting for uncertainty.

3.6.2 Coarse Matching

This component supervises the hybrid coarse registration module, ensuring both the validity of the Mamba encoder and the accuracy of the Transformer interaction. It consists of spot matching loss and coarse matching loss.

Spot Matching Loss $\mathcal{L}_{\mathrm{s}}$. To supervise the layer-wise coarse matching scores $\boldsymbol{P}^{(l)}$, $l \in \{1, \cdots, L\}$, the spot matching loss is adopted:

$$\mathcal{L}_{\mathrm{s}} = -\frac{1}{L}\sum_{l=1}^{L}\frac{1}{\sum_{(i,j)\in\mathcal{C}} o_{ij}}\sum_{(i,j)\in\mathcal{C}} o_{ij}\log \boldsymbol{P}_{ij}^{(l)} \tag{24}$$

where $\mathcal{C}$ is the ground-truth coarse correspondence set with an overlap ratio o_{ij} for each correspondence $(i, j) \in \mathcal{C}$.

Furthermore, when the patch centered at point $\boldsymbol{x} \in \mathbb{R}^3$ is a spherical neighborhood of radius r, the overlapping ratio o_{ij} of the patches centered at $\boldsymbol{p}_i^{\mathcal{S}} \in \mathcal{P}^{\mathcal{S}}$ and $\boldsymbol{q}_j^{\mathcal{S}} \in \mathcal{Q}^{\mathcal{S}}$ with ground-truth rotation $\boldsymbol{R} \in \mathfrak{so}(3)$ and translation $\boldsymbol{t}$ can be calculated as:

$$o_{ij} = \frac{2\pi\int_{D/2}^{r}(r^2 - h^2)\mathrm{d}h}{4\pi r^3/3} = 1 - \frac{3D}{4r} + \frac{D^3}{16r^3} \tag{25}$$

where $D = \max\{\|\boldsymbol{R}\boldsymbol{p}_i^{\mathcal{S}} + \boldsymbol{t} - \boldsymbol{q}_j^{\mathcal{S}}\|, 2r\}$.

Coarse Matching Loss $\mathcal{L}_{\mathrm{c}}$. To supervise the final coarse matching scores $\boldsymbol{P}$, the coarse matching loss can be calculated as:

$$\mathcal{L}_{\mathrm{c}} = -\frac{1}{\sum_{(i,j)\in\mathcal{C}} o_{ij}}\sum_{(i,j)\in\mathcal{C}} o_{ij}\log \boldsymbol{P}_{ij} - \frac{1}{|\mathcal{N}_{\mathcal{P}}|}\sum_{k\in\mathcal{N}_{\mathcal{P}}}\log(1-\hat{o}_k^{\mathcal{P}}) - \frac{1}{|\mathcal{N}_{\mathcal{Q}}|}\sum_{k\in\mathcal{N}_{\mathcal{Q}}}\log(1-\hat{o}_k^{\mathcal{Q}}) \tag{26}$$

where $\mathcal{N}_{\mathcal{P}}$ and $\mathcal{N}_{\mathcal{Q}}$ are sets of semi-dense nodes in point clouds $\mathcal{P}$ and $\mathcal{Q}$ without correspondences.

The total coarse matching loss is defined as $\mathcal{L}_{\mathrm{cm}} = \lambda_{\mathrm{s}}\mathcal{L}_{\mathrm{s}} + \lambda_{\mathrm{c}}\mathcal{L}_{\mathrm{c}}$.

3.6.3 Keypoint Matching

Three losses are employed to supervise similarity calculation, correspondence prediction, and consistency filtering, respectively.

Similarity Calculation Loss $\mathcal{L}_{\mathrm{sc}}$. The InfoNCE loss [30] is adopted to maximize the similarity between the descriptors $d_{\boldsymbol{k}}$ and $d_{\boldsymbol{k}_{\mathrm{gt}}}$ of the true correspondence $(\boldsymbol{k}, \boldsymbol{k}_{\mathrm{gt}})$, while minimizing the similarity between the descriptors $d_{\boldsymbol{k}}$ and $d_{\boldsymbol{k}_{\mathrm{err}}}$ of the false correspondence $(\boldsymbol{k}, \boldsymbol{k}_{\mathrm{err}})$:

$$\mathcal{L}_{\mathrm{sc}} = -\mathbb{E}\left[\log\frac{\mathrm{e}^{d_{\boldsymbol{k}}^{\mathrm{T}}\boldsymbol{W}d_{\boldsymbol{k}_{\mathrm{gt}}}}}{\mathrm{e}^{d_{\boldsymbol{k}}^{\mathrm{T}}\boldsymbol{W}d_{\boldsymbol{k}_{\mathrm{gt}}}} + \sum_{\boldsymbol{k}_{\mathrm{err}}\in C_{\mathrm{err}}}\mathrm{e}^{d_{\boldsymbol{k}}^{\mathrm{T}}\boldsymbol{W}d_{\boldsymbol{k}_{\mathrm{err}}}}}\right] \tag{27}$$

where $\mathbb{E}$ refers to mathematical expectation, C_{err} is a negative sample constraint set defined within the spatial range of the local patch C_{local}, i.e., $C_{\mathrm{err}} \subset C_{\mathrm{include}}$.

Correspondence Prediction Loss $\mathcal{L}_{\mathrm{cp}}$. The L_2 loss supervises the predicted correspondence $\hat{\boldsymbol{q}}$ by minimizing the following expression:

$$\mathcal{L}_{\mathrm{cp}} = \mathbb{E}_{(\boldsymbol{p},\hat{\boldsymbol{q}})}\|\boldsymbol{R}\boldsymbol{p} + \boldsymbol{t} - \hat{\boldsymbol{q}}\|_2 \tag{28}$$

Consistency Filtering Loss $\mathcal{L}_{\mathrm{cf}}$. A binary ground-truth label is defined based on whether the distance is less than the threshold, deciding whether it is an inlier $R_{\mathrm{f}} > 0$, and binary cross-entropy is used to supervise the inlier confidence:

$$\mathcal{L}_{\mathrm{cf}} = \mathrm{BCE}(\mathrm{score}, \mathrm{inlier\ label}) \tag{29}$$

where inlier label = 1 when $\|\boldsymbol{R}\boldsymbol{p} + \boldsymbol{t} - \hat{\boldsymbol{q}}\|_2 < R_f$.

The total keypoint matching loss is $\mathcal{L}_{km} = \lambda_{sc}\mathcal{L}_{sc} + \lambda_{cp}\mathcal{L}_{cp} + \lambda_{cf}\mathcal{L}_{cf}$.

3.6.4 Dense Registration

The dense registration module is supervised using the translation loss $\mathcal{L}_{tr}$ and rotation loss $\mathcal{L}_{rt}$:

$$\mathcal{L}_{tr} = \left\|\hat{\boldsymbol{t}} - \boldsymbol{t}\right\|_2 \tag{30}$$

$$\mathcal{L}_{rt} = \left\|\hat{\boldsymbol{R}}^{\mathrm{T}}\boldsymbol{R} - \boldsymbol{I}\right\|_F \tag{31}$$

where F means Frobenius norm.

The total dense registration loss is defined as $\mathcal{L}_{dr} = \lambda_{tr}\mathcal{L}_{tr} + \lambda_{rt}\mathcal{L}_{rt}$.

4 Experiment

4.1 Datasets and Metrics

4.1.1 Datasets

To evaluate the performance of AMBIR and its advantages over other state-of-the-art methods, the experiments adopt two types of datasets: the indoor point cloud dataset 3DMatch [31], the indoor **low-overlap** point cloud dataset 3DLoMatch [32], as well as the **large-scale** outdoor point cloud datasets KITTI [33].

4.1.2 Metrics

For the indoor datasets 3DMatch and low-overlap 3DLoMatch, the experiments adopt the evaluation metrics as follows:

- **Registration Recall (RR):** Measures the percentage of point cloud pairs successfully aligned within a specified Root Mean Square Error (RMSE < 0.2 m);
- **Inlier Ratio (IR):** Quantifies the proportion of correspondences within a certain residual threshold under the ground-truth transformation;
- **Feature Matching Recall (FMR):** Evaluates the percentage of point cloud pairs with an IR exceeding 5%.

For the outdoor large-scale datasets KITTI, the experiments also adopt the evaluation metrics from Predator [32], namely:

- **Relative Rotation Error (RRE):** The geodesic distance between the estimated and ground-truth rotation matrices;
- **Relative Translation Error (RTE):** The Euclidean distance between the estimated and ground-truth translation vectors;
- **Registration Recall (RR):** Represents the proportion of point cloud pairs where both RRE and RTE are below specific thresholds (RRE < 5° and RTE < 2 m).

4.2 Environment and Parameters

4.2.1 Experimental Environment

For a fair comparison, all models involved in the experiments were executed in the same environment, which was equipped with a 14-core Intel Xeon (R) Platinum 8362 CPU and a single NVIDIA RTX 3090 GPU

with 24 GB of VRAM. All code was compiled on the Linux Ubuntu 22.04 operating system with 32 GB of RAM allocated.

4.2.2 Parameters Setting

AMBIR is trained using the Muon [34] optimizer with a batch size of 1, an initial learning rate of 1×10^{-4}, and a weight decay of 1×10^{-4}. The learning rate scheduler reduces the learning rate to 90% of its previous value every 5 steps. During backpropagation, the gradient norm is clipped to 0.5. The training loss function requires only a single step. Train the model for 5, 5, and 40 epochs on the 3DMatch, 3DLoMatch, and KITTI datasets, respectively. For 3DMatch and 3DLoMatch, the hyperparameters are set to $\lambda_s = 0.1$, $\lambda_c = 1$, $\lambda_{sc} = 1$, $\lambda_{cp} = 10$, $\lambda_{cf} = 1$, $\lambda_{tr} = 5$ and $\lambda_{rt} = 20$. For KITTI, the hyperparameters are set to $\lambda_s = 0.1$, $\lambda_c = 0.2$, $\lambda_{sc} = 1$, $\lambda_{cp} = 1$, $\lambda_{cf} = 1$, $\lambda_{tr} = 5$ and $\lambda_{rt} = 20$. To enhance robustness, RANSAC [35] is employed as a post-processing step to estimate transformations. K is set to 5 uniformly, resulting in 5 training steps.

4.3 Experimental Result

4.3.1 Result on 3DMatch and 3DLoMatch Datasets

As shown in Table 1, 5000, 2500, 1000, 500, and 250 points are sampled from the 3DMatch and 3DLoMatch datasets. Among them, 5000 and 2500 are categorized as dense point clouds, 1000 as medium-density, and 500 and 250 as sparse.

Table 1: Results on 3DMatch dataset.

Method	RR (%) ↑					IR (%) ↑					FMR (%) ↑				
	5000	**2500**	**1000**	**500**	**250**	**5000**	**2500**	**1000**	**500**	**250**	**5000**	**2500**	**1000**	**500**	**250**
CoFiNet [6]	89.3	88.9	88.4	87.4	87.0	49.8	51.2	51.9	52.2	52.2	98.1	98.3	98.1	98.2	98.3
GeoTransformer [7]	92.0	91.8	91.8	91.4	91.2	71.9	75.2	76.0	82.2	85.1	97.9	97.9	97.9	97.9	97.6
OIF-Net [13]	92.4	91.9	91.8	92.1	91.2	62.3	65.2	66.8	67.1	67.5	98.1	98.1	97.9	98.4	98.4
RoITr [14]	91.9	91.7	91.8	91.4	91.0	82.6	82.8	83.0	83.0	83.0	98.0	98.0	97.9	98.0	97.9
PEAL [9]	94.4	94.1	94.1	93.9	93.4	–	–	–	–	–	–	–	–	–	–
SIRA-PCR [15]	93.6	93.9	93.9	92.7	92.4	70.8	78.3	83.7	85.9	87.4	98.2	**98.4**	**98.4**	**98.5**	**98.5**
AMR [10]	94.4	94.3	94.5	94.0	93.9	75.0	81.6	86.3	88.2	89.4	98.3	98.3	98.3	98.3	98.3
CAST [8]	–	–	95.2	–	–	–	–	91.2	91.5	93.1	–	98.3	98.3	98.4	98.3
MT-PCR [20]	–	–	95.5	–	–	–	–	–	–	–	–	–	–	–	–
AMBIR (Ours)	**95.9**	**95.8**	**95.8**	**95.3**	**95.1**	**78.2**	**82.7**	**91.4**	**91.8**	**93.2**	**98.4**	**98.4**	98.3	98.3	98.3

Note: **Bold** font means best.

In the 3DMatch dataset, AMBIR achieves state-of-the-art (sota) performance in both RR and IR across various sampling numbers, while remaining on par with sota levels for FMR. Fig. 6 shows the qualitative registration results on 3DMatch dataset. Compared with the two best open-source SOTA methods, AMR and CAST, both methods exhibit local matching due to similar positions within the red-boxed region, leading to overall misalignment. In contrast, AMBIR does not suffer from this issue.

As shown in Table 2, in the 3DLoMatch dataset, the RR and IR of AMBIR significantly outperform current non-iterative models and achieve performance comparable to the AMR [10] iterative model across most sampling numbers. Regarding FMR, it ranks second only to the leading RoITr [14] model and maintains performance similar to other SOTA models. These results indicate the effectiveness of AMBIR in handling challenging low-overlap registration scenarios.

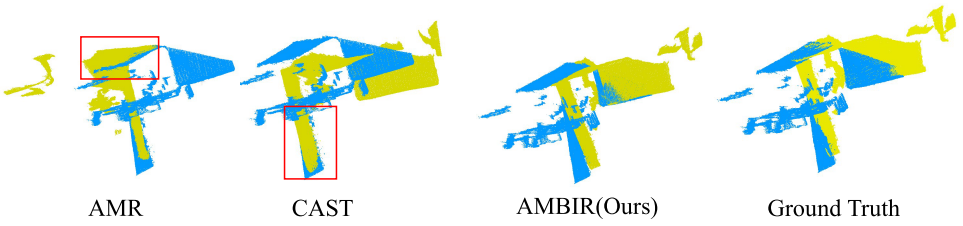

Figure 6: Qualitative registration results on 3DMatch dataset.

Table 2: Results on 3DLoMatch dataset.

Method	RR (%) ↑					IR (%) ↑					FMR (%) ↑				
	5000	2500	1000	500	250	5000	2500	1000	500	250	5000	2500	1000	500	250
CoFiNet [6]	67.5	66.2	64.2	63.1	61.0	24.4	25.0	26.7	26.8	26.9	83.1	83.5	83.3	83.1	82.6
GeoTransformer [7]	75.0	74.8	74.2	74.1	73.5	43.5	45.3	46.2	52.9	57.7	88.3	88.6	88.8	88.6	88.3
OIF-Net [13]	76.1	75.4	75.1	74.4	73.6	27.5	30.0	31.2	32.6	33.1	84.6	85.2	85.5	86.6	87.0
RoITr [14]	74.7	74.8	74.8	74.2	73.6	54.3	54.6	55.1	55.2	55.3	**89.6**	**89.6**	**89.5**	**89.4**	**89.3**
PEAL [9]	79.2	79.0	78.8	78.5	77.9	49.1	54.1	60.5	63.6	65.0	89.1	89.2	89.0	89.0	88.8
SIRA-PCR [15]	73.5	73.9	73.0	73.4	71.1	43.3	49.0	55.9	59.8	60.7	88.8	89.0	88.9	88.6	87.7
AMR [10]	80.0	**80.4**	79.2	78.8	**78.8**	**49.7**	55.4	61.8	64.5	66.2	86.3	85.9	86.0	86.1	85.9
CAST [8]	–	–	75.1	–	–	–	–	**66.3**	66.3	66.5	–	83.1	83.6	85.5	84.7
MT-PCR [20]	–	–	75.4	–	–	–	–	–	–	–	–	–	–	–	–
AMBIR (Ours)	**80.5**	80.3	**79.6**	**79.0**	**78.8**	**49.7**	**58.7**	64.8	**66.5**	**67.3**	86.4	85.8	85.5	85.2	84.8

Note: **Bold** font means best.

4.3.2 Result on KITTI Dataset

As shown in Table 3, AMBIR achieves a 100% RR on KITTI, matching the SOTA performance of recent years. In terms of RTE and RRE, AMBIR outperforms both iterative models-PEAL [9] and AMR [10]-while simultaneously reaching SOTA levels. This demonstrates that AMBIR delivers exceptional performance on large-scale point clouds while remaining highly effective in low-overlap scenarios. Fig. 7 shows the qualitative registration results of AMBIR on the KITTI dataset, where it can be seen that the performance of AMBIR is very close to the ground truth.

Table 3: Results on KITTI dataset.

Method	RTE (cm) ↓	RRE (°) ↓	RR (%) ↑
CoFiNet [6]	8.2	0.41	99.8
GeoTransformer [7]	6.8	0.24	99.8
OIF-Net [13]	6.5	0.23	99.8
PEAL [9]	6.8	0.23	99.8
AMR [10]	6.3	0.23	99.8
RegFormer [16]	8.4	0.24	99.8
CAST [8]	**2.5**	0.27	**100.0**
MT-PCR [20]	2.6	**0.16**	**100.0**
AMBIR (Ours)	3.1	0.20	**100.0**

Note: **Bold** font means best.

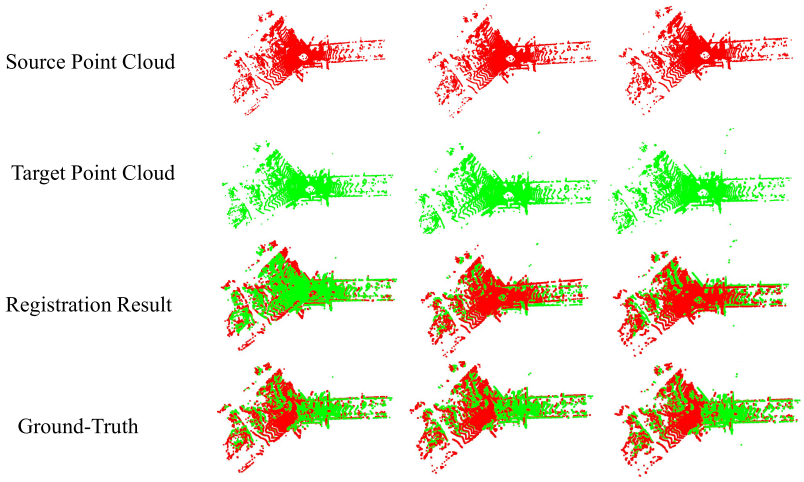

Figure 7: Qualitative registration results of AMBIR on KITTI dataset.

4.3.3 Performance Comparison

Benefiting from the linear attention mechanism of Mamba, as shown in Table 4, AMBIR achieves the shortest average runtime, the lowest VRAM consumption, and a reduced number of FLOPs among all compared methods. It is worth emphasizing that although AMBIR is an iterative approach, the linearization provided by Mamba enables it to achieve, or even surpass, SOTA registration performance while maintaining a relatively small resource footprint. Consequently, by leveraging Mamba modules to minimize computational overhead for large-scale PCR, AMBIR simultaneously remains highly effective for low-overlap registration tasks through its iterative framework.

Table 4: Comparison on performance.

Methods	Average Runtime (s) ↓	VRAM (MB) ↓	FLOPs (G) ↓
GeoTransformer [7]	0.192	12,335	969
RoITr [14]	0.457	15,874	2318
PEAL [9]	2.074	14,529	2851
AMR [10]	1.964	13,768	9883
CAST [8]	0.182	4189	218
MT-PCR [20]	0.178	4091	**129**
AMBIR (Ours)	**0.172**	**3876**	768

Note: **Bold** font means best.

4.3.4 Sensitivity and Stability Analysis of Iteration Rounds

K is set to 5 in Section 4.2.2. To verify the rationality of this value, a sensitivity and stability analysis of the hyperparameter K is added in this section, investigating the influence of $K \in \{3, 4, 5, 6, 7\}$ on various registration metrics. As shown in Table 5, as the number of iterations K increases, both the RR and IR show an upward trend, while the average inference time of the model also rises accordingly. It can be observed that when K increases from 3 to 5, RR and IR improve significantly without incurring significant time overhead. However, when $K > 5$, further increasing K only brings marginal improvements to RR and IR, accompanied by a substantial growth in computational cost, so the practical value of such gains is limited. Therefore, $K = 5$ is a reasonable choice.

Table 5: Sensitivity and stability analysis of iteration rounds.

K	RR (%) ↑	IR (%) ↑	Average Runtime (s) ↓
3	85.5	76.8	0.147
4	92.1	85.3	0.155
5	95.8	91.4	0.172
6	96.0	91.4	0.211
7	96.1	91.6	0.243

4.3.5 Robustness Analysis

To verify the registration robustness of AMBIR under imperfect conditions, tests with point cloud noise and unequal point cloud density are conducted. As shown in Table 6, the integral smoothing property of the Mamba encoder acts as a spatial low-pass filter that suppresses high-frequency noise [36], and the soft-gating mechanism of CAME dynamically assigns low confidence to regions with mismatched geometric densities, preventing sparse artifacts from contaminating global feature aggregation [37]. These mechanisms enable AMBIR to maintain favorable RR values even under severe noise and density variations, providing protection against vulnerabilities under adversarial or noisy conditions.

Table 6: Robustness testing on 3DMatch under noise and unequal density.

Gaussian Noise	RR (%) ↑	Target Density Retention	RR (%) ↑
Clean (σ = 0)	**95.8**	100% (Original)	**95.8**
σ = 0.01 m	94.2	80% Retention	94.8
σ = 0.02 m	91.5	60% Retention	92.1
σ = 0.03 m	87.4	40% Retention	87.5
σ = 0.05 m	83.4	20% Retention	78.2

Note: **Bold** font means best.

4.4 Ablation Studies

4.4.1 Ablation Studies of 3DMatch and KITTI with Comparative Analysis

As shown in Table 7, "w/o" indicates "without," referring to the ablated model lacking the respective module. AMBIR consists of four essential modules: PICOS, Vanilla Mamba, CAME, and Iteration. When evaluating the RR, Average Runtime, and VRAM usage on the 3DMatch, and the RTE, RRE, RR, Average Runtime, and VRAM usage on the KITTI dataset (all with 1000 sampled points). Results show that removing any module degrades the RTE, RRE, and RR metrics. Although removing PICOS saves only a negligible amount of VRAM, it severely compromises registration performance, which is not worthwhile. Notably, while removing the iteration reduces ART and VRAM usage, it is essential for learning overlap priors in low-overlap scenarios, and its removal causes a drastic drop in registration accuracy under such conditions. The above demonstrates that each component is indispensable to the complete AMBIR registration network.

For comparative analysis between indoor and outdoor datasets, the differences in the action mechanisms of each core module under varying scale and scene conditions are elaborated. PICOS ensures consistent semantic alignment across scales. The iterative framework uncovers hidden overlaps in occluded indoor scenes and progressively reduces large translational errors in outdoor scenes. Additionally, the soft-gating of CAME dynamically suppresses repetitive indoor clutter and filters vast featureless outdoor

backgrounds. Crucially, the linear $\mathcal{O}(n)$ Mamba backbone prevents the catastrophic memory explosion of $\mathcal{O}(n^2)$ Transformers in large-scale outdoor tasks, while also moderately improving indoor efficiency.

Table 7: Ablation study on 3DMatch and KITTI.

Method	3DMatch			KITTI				
	RR (%) ↑	ART (s) ↓	VRAM (MB) ↓	RTE (cm) ↓	RRE (°) ↓	RR (%) ↑	ART (s) ↓	VRAM (MB) ↓
AMBIR	**95.8**	0.172	3876	**3.1**	**0.20**	**100.0**	0.178	4512
w/o PICOS	95.2	0.303	3765	5.8	0.35	99.2	0.175	4785
w/o CAME	86.3	0.247	7524	7.5	0.37	98.5	0.269	8457
w/o Vanilla Mamba	94.8	0.452	11,500	3.5	0.24	99.8	0.485	15,194
w/o Iteration	95.0	**0.156**	**3586**	5.2	0.38	98.2	**0.162**	**4210**

Note: **Bold** font means best. ART: Average Runtime.

4.4.2 Ablation Study on Serialization Strategy

To compare the impact of different serialization strategies on registration performance, the AMBIR serialization strategy is replaced, and the effectiveness of various strategies is evaluated on the 3DMatch Dataset, as shown in Table 8. It can be observed that, due to the absence of pre-alignment provided by PICOS, both the Hilbert and Z-order curve methods consume substantial network capacity when learning global rigid invariance from extremely long sequences, which severely degrades RR. The pre-alignment mechanism can leverage GPU parallel computing, requiring only a modest increase in runtime and FLOPs.

Table 8: Ablation study on serialization strategy.

Strategy	Average Runtime (s) ↓	FLOPs (G) ↓	RR (%) ↑
Hilbert	**0.165**	**742**	91.0
Z-order	0.179	804	92.7
PICOS (Ours)	0.172	768	**95.8**

Note: **Bold** font means best.

5 Conclusion and Future Work

To address the challenges that current Transformer-based PCR frameworks suffer from quadratic computational complexity, leading to excessive resource consumption in large-scale scenarios and suboptimal performance in low-overlap environments, an iterative PCR network, AMBIR, fusing Attention and Mamba, is proposed. Specifically, an iterative network architecture is incorporated into the backbone to learn overlap information from prior registration results, thereby enhancing registration performance by leveraging knowledge from the preceding step. To convert 3-D point cloud data to linear data for the Mamba encoder, Prior-Informed Co-aligned Serialization is proposed to ensure that points with adjacent indices after serialization are spatial neighbors, thereby improving the efficiency and robustness of the subsequent registration process. After that, a Consistency-Aware Mamba Encoder is introduced to leverage its advantage in linear computational complexity, making the method more suitable for large-scale point clouds. Overall, AMBIR integrates the advantages of iterative networks in low-overlap scenarios with the benefits of the linear complexity of the Mamba model. It simultaneously resolves the PCR challenges in both scenarios, achieving a balance between performance and efficiency.

Future work will apply AMBIR to industrial and scientific instrument tasks, such as deformation monitoring of large astronomical telescope surfaces, to further broaden its scope of applications. In addition, investigating noise-resistant models under artificial or extreme conditions, as well as mechanisms

to prevent noise-induced errors from propagating and amplifying across iterations, is also a worthwhile research direction.

Acknowledgement: None.

Funding Statement: This work was supported by the National Natural Science Foundation of China (Grant No. 12141304).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Haotian Cao; methodology, Haotian Cao; software, Haotian Cao; validation, Haotian Cao; formal analysis, Haotian Cao; investigation, Haotian Cao; resources, Qingsheng Zhu; data curation, Haotian Cao; writing—original draft preparation, Haotian Cao and Qingsheng Zhu; writing—review and editing, Haotian Cao and Qingsheng Zhu; visualization, Haotian Cao; supervision, Qingsheng Zhu; project administration, Qingsheng Zhu; funding acquisition, Qingsheng Zhu. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The 3DMatch Dataset used in this study is publicly available at https://3dmatch.cs.princeton.edu (accessed on 6 March 2026). The KITTI Dataset used in this study is publicly available at https://www.cvlibs.net/datasets/kitti (accessed on 6 March 2026). The source code and model weights of the study are available from the authors upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Pomerleau F, Colas F, Siegwart R. A review of point cloud registration algorithms for mobile robotics. Found Trends Robot. 2015;4(1):1–104.
2. Tao Y, Yang X, Wang H, Wang J, Li Z, Liang H. Lsreg-net: an end-to-end registration network for large-scale lidar point cloud in autonomous driving. IEEE Sens J. 2025;25(11):20675–86. doi:10.1109/jsen.2025.3562916.
3. Zhang YX, Gui J, Yu B, Cong X, Gong X, Tao W, et al. Deep learning-based point cloud registration: a comprehensive survey and taxonomy. arXiv:2404.13830. 2024.
4. Besl PJ, McKay ND. A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell. 1992;14(2):239–56.
5. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems; 2017 Dec 4–9; Long Beach, CA, USA. p. 6000–10.
6. Yu H, Li F, Saleh M, Busam B, Ilic S. Cofinet: reliable coarse-to-fine correspondences for robust pointcloud registration. In: Proceedings of the 35th International Conference on Neural Information Processing Systems; 2021 Dec 6–14; Online. p. 23872–84.
7. Qin Z, Yu H, Wang C, Guo Y, Peng Y, Xu K. Geometric transformer for fast and robust point cloud registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022 Jun 18–24; New Orleans, LA, USA. p. 11143–52.
8. Huang R, Tang Y, Chen J, Li L. A consistency-aware spot-guided transformer for versatile and hierarchical point cloud registration. In: Proceedings of the 38th International Conference on Neural Information Processing Systems; 2024 Dec 10–15; Vancouver, BC, Canada. p. 70230–58.
9. Yu J, Ren L, Zhang Y, Zhou W, Lin L, Dai G. PEAL: prior-embedded explicit attention learning for low-overlap point cloud registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023 Jun 17–24; Vancouver, BC, Canada. p. 17702–11.
10. Chen Z, Ren Y, Zhang T, Dang Z, Tao W, Susstrunk S, et al. Adaptive multi-step refinement network for robust point cloud registration. arXiv:2312.03053. 2023.

11. Kalman RE. A new approach to linear filtering and prediction problems. J Basic Eng. 1960;82(1):35–45. doi:10.1115/1.3662552.
12. Gu A, Dao T. Mamba: linear-time sequence modeling with selective state spaces. arXiv:2312.00752. 2023.
13. Yang F, Guo L, Chen Z, Tao W. One-inlier is first: towards efficient position encoding for point cloud registration. Adv Neural Inf Process Syst. 2022;35:6982–95.
14. Yu H, Qin Z, Hou J, Saleh M, Li D, Busam B, et al. Rotation-invariant transformer for point cloud matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023 Jun 17–24; Vancouver, BC, Canada. p. 5384–93.
15. Chen S, Xu H, Li R, Liu G, Fu CW, Liu S. SIRA-PCR: sim-to-real adaptation for 3d point cloud registration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2023 Oct 1–6; Paris, France. p. 14394–405.
16. Liu J, Wang G, Liu Z, Jiang C, Pollefeys M, Wang H. Regformer: an efficient projection-aware transformer network for large-scale point cloud registration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2023 Oct 1–6; Paris, France. p. 8451–60.
17. Xie Y, Wang B, Li S, Zhu J. Iterative feedback network for unsupervised point cloud registration. IEEE Robot Autom Lett. 2024;9(3):2327–34. doi:10.1109/lra.2024.3355784.
18. Ding Y, Li K, Zhang G, Zhu Z, Wang P, Wang Z, et al. Multi-step depth enhancement refine network with multi-view stereo. PLoS One. 2025;20(2):1–17. doi:10.1371/journal.pone.0314418.
19. Wen X, Xiang P, Han Z, Cao YP, Wan P, Zheng W, et al. PMP-Net: point cloud completion by learning multi-step point moving paths. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021 Jun 20–25; Nashville, TN, USA. p. 7443–52.
20. Liu B, Liu A, Chen H, Cui J, Wang Y, Zhang H. MT-PCR: a hybrid mamba-transformer with spatial serialization for hierarchical point cloud registration. arXiv:2506.13183. 2025.
21. Chen C, Li K, Xing K, Wang Y. E2MNet: an end-to-end large-scale point cloud registration network based on Mamba. J Electron Imaging. 2025;34(3):033045. doi:10.1117/1.jei.34.3.033045.
22. Sun Y, Zhang L. MaGo-I2P: image-to-point cloud registration with mamba and geometry recovery. In: Proceedings of the 2025 International Conference on Multimedia Retrieval; 2025 Jun 30–Jul 3; Chicago, IL, USA. p. 1237–45.
23. Li Q, Jiang Y, Cheng J, Chen W, Zhao P, Qiao X, et al. AeroMamba: an efficient mamba-based approach for large-scale point cloud registration in aircraft assembly. In: Proceedings of the 2025 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM); 2025 Jul 14–18; Hangzhou, China. p. 1–8.
24. Alawieh A, Condurache AP. FA-KPConv: introducing euclidean symmetries to KPConv via frame averaging. arXiv:2505.04485. 2025.
25. Katharopoulos A, Vyas A, Pappas N, Fleuret F. Transformers are RNNs: fast autoregressive transformers with linear attention. In: Proceedings of the 2020 12th International Conference on Machine Learning; 2020 Feb 15–17; Shenzhen, China. p. 5156–65.
26. Qi CR, Su H, Mo K, Guibas LJ. Pointnet: deep learning on point sets for 3d classification and segmentation. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition; 2017 Jul 21–26; Honolulu, HI, USA. p. 652–60.
27. Chollet F. Xception: deep learning with depthwise separable convolutions. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition; 2017 Jul 21–26; Honolulu, HI, USA. p. 1251–8.
28. Hendrycks D. Gaussian error linear units (Gelus). arXiv:1606.08415. 2016.
29. Li J, Lee GH. USIP: unsupervised stable interest point detection from 3d point clouds. In: Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision; 2019 Oct 27–Nov 2; Seoul, Republic of Korea. p. 361–70.
30. Oord A, Li Y, Vinyals O. Representation learning with contrastive predictive coding. arXiv:1807.03748. 2018.
31. Zeng A, Song S, Nießner M, Fisher M, Xiao J, Funkhouser T. 3DMatch: learning local geometric descriptors from RGB-D reconstructions. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition; 2017 Jul 21–26; Honolulu, HI, USA. p. 1802–11.

32. Huang S, Gojcic Z, Usvyatsov M, Wieser A, Schindler K. Predator: registration of 3D point clouds with low overlap. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021 Jun 20–25; Nashville, TN, USA. p. 4267–76.
33. Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The kitti vision benchmark suite. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition; 2012 Jun 16–21; Providence, RI, USA. p. 3354–61.
34. Jordan K, Jin Y, Boza V, You J, Cesista F, Newhouse L, et al. Muon: an optimizer for hidden layers in neural networks. 2024 [cited 2024 Dec 20]. Available from: https://github.com/KellerJordan/muon.
35. Fischler MA, Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM. 1981;24(6):381–95. doi:10.1145/358669.358692.
36. Wu Z, Duan Y, Wang H, Fan Q, Guibas LJ. If-defense: 3d adversarial point cloud defense via implicit function based restoration. arXiv:2010.05272. 2020.
37. Yang H, Shi J, Carlone L. Teaser: fast and certifiable point cloud registration. IEEE Trans Robot. 2020;37(2):314–33.