Abstract—Learning robust feature matching between the template and search area is crucial for 3-D Siamese tracking. The core of Siamese feature matching is how to assign high feature similarity to the corresponding points between the template and the search area for precise object localization. In this article, we propose a novel point cloud registration-driven Siamese tracking framework, with the intuition that spatially aligned corresponding points (via 3-D registration) tend to achieve consistent feature representations. Specifically, our method consists of two modules, including a tracking-specific nonlocal registration (TSNR) module and a registration-aided Sinkhorn template-feature aggregation module. The registration module targets the precise spatial alignment between the template and the search area. The tracking-specific spatial distance constraint is proposed to refine the cross-attention weights in the nonlocal module for discriminative feature learning. Then, we use the weighted singular value decomposition (SVD) to compute the rigid transformation between the template and the search area and align them to achieve the desired spatially aligned corresponding points. For the feature aggregation model, we formulate the feature matching between the transformed template and the search area as an optimal transport problem and utilize the Sinkhorn optimization to search for the outlier-robust matching solution. Also, a registration-aided spatial distance map is built to improve the matching robustness in indistinguishable regions (e.g., smooth surfaces). Finally, guided by the obtained feature matching map, we aggregate the target information from the template into the search area to construct the target-specific feature, which is then fed into a CenterPoint-like detection head for object localization. Extensive experiments on KITTI, NuScenes, and Waymo datasets verify the effectiveness of our proposed method.

Index Terms—3-D Siamese tracking, 3-D single object tracking, point cloud registration, robust feature matching.

I. INTRODUCTION

VISUAL object tracking serves as the key component in a variety of computer vision applications, such as autonomous driving [1], robot vision [2], and augmented reality [3]. With the development of inexpensive LiDAR sensors, more research has turned to 3-D object tracking at the point cloud level and achieved substantial progress. In general, given the annotated 3-D bounding box (BBox) of the target in the first frame, the tracker aims to predict the positions of the target in the remaining scanning frames. Compared to trackers using 2-D images [4], [5], point cloud-based trackers are inherently more robust in challenging situations, for example, illumination and weather changes. However, reliable 3-D object tracking in the real world remains a challenging problem in the presence of sparse scenes, object occlusion, and LiDAR noise.

In recent years, 3-D Siamese object tracking has attracted increasing research interest due to the success of 2-D Siamese tracking and the development of deep learning in the 3-D vision field. The core of the Siamese tracker is to obtain reliable feature matching [6], [7], [8] between the target template and the search area, thereby robustly distinguishing the target from the search area. To this end, many efforts have been made toward improving the robustness of feature matching in the Siamese tracking field. For example, SC3D [9] develops a shape completion-enhanced Siamese network to enrich the object semantics of point features for improving the matching quality. BAT [10] uses BoxCloud representations to capture size-aware and part-aware object clues for matching enhancement. P2B [11] and V2B [12] rely on discriminative point feature representations using a hierarchical feature aggregation mechanism in PointNet++ [13] for high-quality feature matching. However, despite their effectiveness, all of these trackers overlook the feature differences caused by the spatial misalignment between corresponding points. A significant positional deviation can reduce the feature consistency of corresponding points, potentially degrading the feature-matching robustness.

Based on the analysis above, this article proposes a simple, yet effective 3-D Siamese tracking framework that leverages point cloud registration to mitigate the feature inconsistency caused by spatial misalignment for robust feature matching. The motivation behind this approach is that the Siamese feature extraction backbone is essentially a projection function from 3-D coordinate space to feature space.
Therefore, by spatially aligning corresponding points between templates and search areas using 3-D registration techniques, the aligned corresponding points can own (nearly) consistent spatial coordinates, resulting in more consistent feature representations and thus improving matching robustness (see Fig. 1).

Specifically, our framework consists of two modules, including a tracking-specific nonlocal registration (TSNR) module and a registration-aided Sinkhorn feature aggregation (RSFA) module. Taking as inputs the template and the search area, the registration module aims to accurately predict the rigid transformation for their spatial alignment by alternately performing the nonlocal feature extraction, the inlier classification, and the weighted singular value decomposition (SVD). To achieve a discriminative feature representation, TSNR exploits the tracking-specific spatial distance constraint between the corresponding points to refine the cross-attention weights in the nonlocal module. The intuition of such a spatial constraint is that the distance between the corresponding points from two consecutive frames is usually limited due to the small scanning interval of LiDAR. After transforming the template with the estimated rigid transformation, a PointNet++ backbone [13] is employed to extract the feature embeddings of the transformed template and the search area. Different from previous Siamese trackers that directly use the Cosine distance to measure the feature similarity, RSFA formulates the feature matching (between the transformed template and the search area) as an optimal transport problem and utilizes the Sinkhorn optimization to search for the outlier-robust matching solution. Furthermore, to improve the robustness of the feature matching in indistinguishable regions (e.g., car, door surfaces), a registration-aided spatial distance map is also built to regularize the feature-matching similarity. Consequently, we exploit the refined feature-matching map to guide the target information (defined in the template) aggregation into the search area and form the target-specific search-area feature for object localization via a CenterPoint-like detection head [12], [14]. Extensive experiments on KITTI, NuScenes, and Waymo benchmark datasets verify the effectiveness of our proposed method. To summarize, our main contributions are listed as follows.

1) We propose a novel 3-D registration-driven Siamese tracking framework, where robust feature matching can be obtained by spatially aligning the corresponding points between the template and the search area.
2) We design an effective TSNR module for reliable registration between the template and the search area during 3-D tracking.
3) We propose a novel registration-aided feature aggregation module, where the registration-based spatial distance map is proposed for feature-matching refinement.
4) Compared to current state-of-the-art (SOTA) methods, our proposed tracker can obtain outstanding performance on extensive benchmark datasets.

II. RELATED WORK

A. Three-Dimensional Single Object Tracking

Due to the inherent robustness of point clouds to real-world tracking challenges such as illumination changes and less-textured scenes, more research has moved from RGB-based 2-D tracking to 3-D tracking that primarily utilizes LiDAR-scanned point clouds for object tracking. SC3D [9] pioneers to leverage shape completion to enhance the feature representation of the template and the target candidates (obtained by Kalman filtering) and selects the target candidate with the largest feature similarity to the template as the target localization. P2B [11] constructs the discriminative target-specific feature for object identification by integrating the template information into the search area. Then, VoteNet [15] is employed to determine the object position in the search area. 3DSiamRPN [16] proposes to exploit the region proposal network to obtain the proposal and scores for object localization. BAT [10] proposes to construct the BBox-aware feature descriptor for a more robust feature matching. In addition, to handle the tracking task in the sparse point cloud scenes, V2B [12] utilizes the discriminative shape-aware template embedding for feature aggregation and replaces the VoteNet with a voxel-to-BEV detector for target location. Recently, inspired by the success of Transformers, PTT [17] employs the Transformer-based self-attention mechanism for feature learning. PTTR [18] fuses the self-attention and cross-attention mechanisms in Transformers for robust target-specific feature learning and achieves impressive tracking accuracy. Moreover, CAT [19] mines rich spatial and temporal contextual information from the LiDAR sequence to enhance the tracking performance.

B. Three-Dimensional Multiobject Tracking

Most multobject trackers such as [20], [21], [22] first exploit a 3-D detector [23], [24] to determine the targets to be tracked in all frames (i.e., tracking-by-detection). Then, the data association among the detected objects in the consecutive frames is used for object trajectory prediction. Weng and Kitani [25] proposed to utilize the 3-D Kalman filter for tracking state estimation and then exploit the Hungarian algorithm for matching the detected objects. Wang et al. [26] further used a GNN module for the object relationship modeling and
focused on jointly optimizing the object detection and the data association. CenterPoint [14] finds the centers of objects and regresses other tracking properties using the keypoint detector and further refines these estimated object locations via additional point features based on the predicted position. Other methods [20], [27] also show enlightening results.

C. Point Cloud Registration

Taking as input a pair of source and target point clouds, the point cloud registration aims to recover their potential rigid transformation for spatial alignment. For traditional registration methods, the iterative closest point (ICP) algorithm [28] iteratively performs correspondence estimation and least-squares optimization for transformation estimation. Other ICP variants, such as Go-ICP [29] and Trimmed ICP [30], improve ICP in terms of the local optima problem and the partially overlapping registration. For learning-based methods, DCP [31] utilizes feature similarity maps to generate pseudo-correspondences and then computes the rigid geometric characteristic and the deep feature, IDAM [33] in a self-supervised manner. Furthermore, by fusing the keypoint detector and further refines these estimated object locations via additional point features based on the predicted position.

III. APPROACH

A. Problem Setting

In point cloud-based single object tracking, given the 3-D BBox $b_1$ of the target in the first frame, the tracker aims to predict the BBoxes $\{b_i\}$ of the target in remaining scanning frames. The BBox can be parameterized with seven elements, including the object center coordinate $(x, y, z)$, the object size $(l, w, h)$, and the heading angle $\theta$ (we assume the rotation of the object is just around the z-axis). Since the size of BBox remains consistent with $b_1$ during tracking, we need not infer the object size $(l, w, h)$. Following the Siamese network paradigm, we aim to localize the target (defined by the template) in the search area frame by frame. Specifically, in frame $t$, we first construct the template $X = \{x_i \in \mathbb{R}^3 | i = 1, \ldots, N\}$ and search area $Y = \{y_j \in \mathbb{R}^3 | j = 1, \ldots, M\}$ by cropping frame $t - 1$ with $b_{t-1}$ and frame $t$ with enlarged $b_{t-1}$, respectively. Then, taking as input the template and the search area, the Siamese network is parameterized with $b_1$ during training, we need not infer the object size $(l, w, h)$.

B. Tracking-Specific Nonlocal Registration

The TSNR module formulates the template and search area $(X, Y)$ as the partially overlapped source and target point cloud pair in the 3-D registration paradigm and aims to align $X$ to $Y$ by inferring their rigid transformation consisting of a rotation matrix $R \in SO(3)$ and a translation vector $t \in \mathbb{R}^3$.

1) Tracking-Specific Nonlocal Feature Learning

The feature learning network used in our registration module is the tracking-specific iterative nonlocal module (TSNonlocal), which receives the template and the search area as inputs and produces a geometric feature for each point. To realize robust registration performance, our TSNonlocal module explicitly
utilizes the spatial distance constraint between the corresponding points of the template and the search area to learn a discriminative feature representation. As illustrated in Fig. 3, TSNonlocal contains $T$ iterations for feature updating, and the initial feature embeddings $\{0\}F_x \in \mathbb{R}^d$ and $\{0\}F_y \in \mathbb{R}^d$ are obtained by mapping each point $x_i$ and $y_j$ into a common feature space with a multilayer perception (MLP). Then, the initial features are iteratively enhanced through the nonlocal message passing from the search area to the template $(Y \rightarrow X)$ and the template to the search area $(X \rightarrow Y)$. To avoid repeating, we just describe the message passing about $X \rightarrow Y$. Specifically, in the $r$th iteration ($1 \leq r \leq T$), the enhanced point feature $\{r\}F_{x_i} \in \mathbb{R}^d$ is obtained by the way that the query $\{r\}Q_{x_i} = \{r\}W_{q}(\{r-1\}F_{x_i}) \in \mathbb{R}^d$ retrieves the value $\{r\}V_{y_j} = \{r\}W_{v}(\{r-1\}F_{y_j}) \in \mathbb{R}^d$ with key $\{r\}K_{x_i} = \{r\}W_{k}(\{r-1\}F_{x_i}) \in \mathbb{R}^d$ ($\{r\}W_{q}$ and $\{r\}W_{v}$ are the learnable parameters)

$$\quad \{r\}F_{x_i} = \{r-1\}F_{x_i} + \text{MLP}\left(\sum_{j=1}^{M} \alpha_{i,j} \beta_{i,j} \{r\}V_{y_j}\right)$$

(1)

where the cross-attention weight $\alpha_{i,j}$ is defined as the dot-product similarity between the query and the key

$$\quad \alpha_{i,j} = \text{softmax}\left(\frac{Q_{x_i}K_{y_j}}{\sqrt{d}}\right).$$

(2)

Furthermore, we exploit a spatial distance constraint $\beta_{i,j}$ between the template and the search area to regularize the cross-attention weight. The intuition of this spatial constraint lies in the observation that the distance between the corresponding points of the template and the search area is limited due to small LiDAR scanning interval and the relatively stable object motion. As depicted in Fig. 3, the corresponding point of template point $A$ tends to lie in its spatially close points $(D, F, C)$ in the search area, while the far points $(B, E)$ may be rejected. This spatial constraint-enhanced cross-attention explicitly mitigates the unexpected message passing from the noncorresponding points $(B, E)$ to $A$, thereby assisting more robust message passing and feature learning. In our implementation, we heuristically design the spatial distance constraint as below

$$\beta_{ij} = \min\{d_{ij} \leq 1\}, \quad d_{ij} = \left(\frac{y_{j,y} - x_{i,y}}{a^2} + \frac{y_{j,z} - x_{i,z}}{b^2}\right)$$

(3)

where $x_{i,y}$ and $x_{i,z}$ denote the $y$-axis value (left/right direction) and the $z$-axis value (up/down direction) of template point $x_i$, respectively. For each template-point $x_i$, its possible corresponding points in the search area are defined as the points whose Euclidean distances to $x_i$ fall in an ellipse area $(a$ and $b$ are the semimajor and semiminor axes, respectively) in the $yz$-plane $(x$-axis denotes the forward/backward direction of the object). Notably, choosing the ellipse area $(a > b)$ of the $yz$-plane is due to that for moving objects on the ground, the motion along the $y$-axis tends to be greater than along the $z$-axis. For simplicity, we denote the learned point features $\{t\}F_{x_i}$ and $\{t\}F_{y_j}$ after the last iteration as $F_{x_i} = \{F_{x_i}\}$ and $F_{y_j} = \{F_{y_j}\}$.

2) Inlier Classifier for Outlier Rejection: Due to the outlier interference (caused by the LiDAR noise or background points in the search area), it is difficult to directly predict the precise rigid transformation between the original template and the search area. To relieve it, we construct an inlier classifier built on a three-layer MLP for outlier filtering on both the template and the search area. Specifically, with the learned point feature as input, the classifier predicts the inlier probabilities $\{c_{x_i}\}$ and $\{c_{y_j}\}$ for points $x_i$ and $y_j$. Then, we choose $k$ points with the highest inlier probabilities as the inliers $X = \{x_i | c_{x_i} \geq t_x\}$ and $Y = \{y_j | c_{y_j} \geq t_y\}$, where $t_x$ and $t_y$ denote the top-$k$ probability thresholds. We denote the number of points in the filtered template and the search area as $N' = |X|$ and $M' = |Y|$, respectively.

After rejecting the outliers, a matching map $M \in \mathbb{R}^{N' \times M'}$ is established with a softmax function to generate the soft correspondence $\hat{y}_j$ for each template point $x_i'$

$$\hat{y}_j = \sum_{j=1}^{M'} M_{i,j} y_j', \quad M_{i,j} = \text{softmax}\left(\left[\{F_{x_i'}F_{y_j'}, \ldots, F_{x_i'}F_{y_{j'}}\}\right]_j\right)$$

(4)

Finally, we use the weighted SVD to solve the least-square fitting over the generated soft correspondence pairs $(x'_i, \hat{y}_j)$ for rigid transformation estimation

$$\hat{R}, \hat{t} = \arg \min_{R,t} \sum_{i=1}^{N'} c_{x_i} \|R x'_i + t - \hat{y}_j\|_2$$

(5)

where the inlier probability controls the optimization weight on each least-square item. Consequently, we align the template...
to the search area with the predicted rigid transformation \([\bar{R}, \hat{t}]\) and utilize the transformed template \(\bar{X} = [\bar{x}_i | \bar{R} \bar{x}_i + \hat{t}]\) and search area \(Y\) for Siamese object localization. We note that the spatially aligned corresponding points via 3-D registration tend to achieve consistent feature representation, thereby improving the robustness of the feature matching as shown in Fig 1.

3) Discussion About Tracking Just Using Registration: Although our proposed tracking-specific registration module presents high registration precision between the consecutive frames, it should be noted that the only-registration-based tracker tends to suffer from serious error accumulation issues, resulting in tracking failure. For example, assume that there is a BBox error \(\Delta_t\) between the predicted BBox \(\bar{b}\) and the ground-truth BBox \(b\), in frame \(t\) (i.e., \(b = b + \Delta_t\)). Then, even if we estimate the perfect relative pose \(T^t\) between frame \(t\) and frame \(t + 1\), the transformed BBox \(\bar{b}_{t+1} = T^t(b)\) still encounters the same BBox error \(\Delta_t\) with ground-truth BBox \(b_{t+1}\) (i.e., \(b_{t+1} = b_{t+1} + \Delta_t\)). Therefore, in the case of imperfect relative pose estimation, the accumulated error tends to increase gradually as the tracking proceeds, causing poor tracking accuracy.

Based on the discussion above, we still need the feature aggregation and object localization for 3-D tracking. The feature aggregation aims to leverage feature matching to guide the transfer of target clues from the template into the search area. As such, the search area embedded with the target information can be used to predict the probable position of the object. This implies that, despite the existence of some errors for the predicted BBox in the previous frame, the search area gathering target cues from the template is still capable of providing a more precise object localization in the current frame for solving the error accumulation.

C. Registration-Aided Sinkhorn Feature Aggregation

The RSFA module aims to integrate the target information (defined in the transformed template) into the search area for object localization and mainly contains two components: the Sinkhorn optimization-based feature matching and the registration-aided matching refinement.

1) Sinkhorn Optimization-Based Feature Matching: Following previous Siamese methods, we extract the point-wise features \(\Phi_X = \{\Phi_{x_i} \in \mathbb{R}^d\}\) and \(\Phi_Y = \{\Phi_{y_j} \in \mathbb{R}^d\}\) of the transformed template and the search area using PointNet++ backbone [13]. Different from the previous methods using Cosine distance to measure the feature similarity, we formulate the feature matching as the optimal transport problem and use the Sinkhorn algorithm [40] to search for the reliable similarity assignment, which is verified to be more robust to the outlier interference [41], [42]. Specifically, we first construct the initial feature matching matrix \(\hat{A} \in \mathbb{R}^{N \times M}\) via feature inner product: \(A_{i,j} = < \Phi_{x_i}, \Phi_{y_j} >\). Since Sinkhorn optimization needs matrix elements to be positive within finite values, we transform the initial matrix via instance normalization and concurrently use an exponential function to map the normalized matrix elements to be positive: \(A_{i,j}^{\text{pos}} = \text{Exp}(\text{Ins. Norm}(A))\). Then, the Sinkhorn algorithm expands the matrix \(A_{i,j}^{\text{pos}}\) with a slack row and a slack column to form a slack-matching matrix

\[ A' = \begin{bmatrix} A_{i,j}^{\text{pos}} & z_1 \\ \bar{z}_2' & z \end{bmatrix}, \quad A' \in \mathbb{R}^{(N+1) \times (M+1)} \]

where \(z_1 \in \mathbb{R}^{M \times 1}, z_2 \in \mathbb{R}^{N \times 1}, z \in \mathbb{R}\) are all optimizable parameters (initialized with zero), providing a similarity assignment space for outliers without corresponding points. After repeatedly alternating the row and column normalizations on \(A'\), we remove the slack variables and utilize the resulting matrix \(\hat{A} \in \mathbb{R}^{N \times M}\) as the feature-matching matrix.

2) Registration-Aided Matching Refinement: Although the spatial alignment and the Sinkhorn algorithm can effectively improve the matching robustness as above, the feature matching may inevitably suffer from the ambiguous similarity assignment on smooth surfaces that lacks significant geometric characteristic (such as the car-door surface). For example, while search-area point \(y_j\) on the car-door surface has a high similarity with its aligned corresponding point \(x_i\), it may also achieve a certain similarity with another template point \(\tilde{x}_k\) also lying on this indistinguishable surface, which potentially degrades the discrimination of \(\hat{A}\). Based on our registration operation, we focus on exploiting the spatial distance constraint of the aligned corresponding points [termed registration matching (RM)] to handle this issue. Specifically, we first construct a spatial distance-based matching matrix \(D_{\text{reg}} \in \mathbb{R}^{N \times M}\) between \(X\) and \(Y\) and then utilize \(D_{\text{reg}}\) to regularize \(\hat{A}\) using Hadamard product

\[ \hat{A}_{i,j}^{\text{reg}} = \hat{A}_{i,j} \cdot D_{i,j}, \quad D_{i,j} = \max\left(1 - \frac{d^2_{i,j}}{\sigma^2}, 0\right), \quad d_{i,j} = \|\bar{x}_i - y_j\|_2 \]

where the parameter \(\sigma\) controls the maximum allowance spatial distance of the corresponding points. If distance \(d_{i,j}\) exceeds the threshold \(\sigma\), the entity \(D_{i,j}^{\text{reg}}\) is clipped to 0. Otherwise, the entity \(D_{i,j}^{\text{reg}}\) is negatively related to \(d_{i,j}\).

Compared to the matching matrix \(\hat{A}\) just at the feature level, the registration-aided spatial distance constraint can effectively relieve the proposed ambiguous matching problem. We continue the example presented above to illustrate it. After registration, the aligned template point \(\tilde{x}_1\) tends to own a much smaller spatial distance with \(y_1\) than \(\tilde{x}_2\). Therefore, we can utilize this spatial distance-based matching score to reweight the feature matching score in \(\hat{A}\) so that the incorrect feature similarity between the noncorresponding pair can be effectively reduced.

3) Target-Specific Feature Aggregation: After establishing the feature-matching matrix \(\hat{A}_{\text{reg}}\), we use it to guide the transferring of the target information (defined in the transformed template) into the search area for target-specific search-area feature learning. Specifically, we fuse the global target information and the local target information into each search-area point \(y_j\). For the global target information, we utilize the matching scores to weigh all template features, on which a max-pooling operation is concurrently performed to achieve the global target information. An MLP is then used to produce that global target embedding: \(F_{\text{y}}^{\text{reg}} = \text{MLP}(\text{MaxPool}(\hat{A}_{\text{reg}}^{i,j} \cdot \Phi_{y_j}))\). For the local target information, we transfer the local...
target information of the most relevant template point $\hat{x}_k^i$ (index $k^* = \arg \max_i \hat{A}_{reg}^{ij}$) into $y_j$. The local target information of template-point $\hat{x}_k^i$, consists of the point feature $\Phi_{\hat{x}_k^i}$, point coordinate $\hat{x}_k$, and the similarity score $\hat{A}_{reg}^{ij}$. These local target information (combining the search-area feature $\Phi_{y_j}$) is then passed into an MLP to build the local target embedding: $F_{ij}^y = MLP(\Phi_{y_j}, \hat{A}_{reg}^{ij}, \Phi_{\hat{x}_k^i})$. Finally, with the concatenated global and local target embeddings, we employ an MLP to fuse them and achieve the desired target-specific feature for point $y_j$: $F_{ij}^{y*} = MLP([F_{ij}^g, F_{ij}^y])$. Finally, we feed the target-specific features $\{F_{ij}^{y*}\}$ in a CenterPoint-like detection head [12], [14] for object center and rotation-angle regression.

### D. Loss Function

The loss functions of our method consist of the registration-level loss and the tracking-level loss. Given the template $X$ with BBox $(x_1, y_1, z_1, \theta_1)$ and search area $Y$ with BBox $(x_2, y_2, z_2, \theta_2)$, we construct the ground-truth rotation matrix $R^*$ and translation vector $t^*$ for supervision

$$R^* = \begin{bmatrix} \cos(\Delta \theta) & -\sin(\Delta \theta) & 0 \\ \sin(\Delta \theta) & \cos(\Delta \theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad t^* = [\Delta x, \Delta y, \Delta z]$$

where $\Delta \theta = \theta_2 - \theta_1$ and $\Delta x = x_2 - x_1$. The ground-truth transformed template is denoted as $\hat{X} = \{\hat{x}_i \in \mathbb{R}^3 | \hat{x}_i = R^*x_i + t^*, x_i \in X\}$.

1) **Inlier Classification Supervision**: To train our inlier classifier, we need to prepare the inlier/outlier labels for all template and search-area points (denoted as $\{c^{x}_i\}$ and $\{c^{y}_j\}$) which are generated via

$$c^{x}_i = \mathbb{I}\{d_{\hat{x}_i \rightarrow y} < \tau\}, \quad c^{y}_j = \mathbb{I}\{d_{\hat{y}_j \rightarrow x} < \tau\}$$

where $d_{\hat{x}_i \rightarrow y} = \min_j \|\hat{x}_i - y_j\|_2$ denotes the minimum $l_2$ distance from $\hat{x}_i$ to $Y$ and $\tau$ is the inlier threshold (we set $\tau = 0.1$ for all our experiments). We adopt the binary cross-entropy (BCE) loss as the supervision signal

$$L_{\text{cls}} = \sum_{i=1} BCE(\hat{c}_x, c^{x}_i) + \sum_{j=1} BCE(\hat{c}_y, c^{y}_j)$$

where $BCE(p, q) = -q \log(p) - (1 - q) \log(1 - p)$.

2) **Correspondence Supervision**: The correspondence supervision aims to improve the quality of the generated soft correspondence $\{\hat{y}_j\}$ for inlier-filtered template points $\{\hat{x}_i\}$ (see Section III-B2) in our registration module. We realize it by minimizing the $l_2$ loss between the soft correspondence and the ground-truth transformed template point as below

$$L_{\text{corr}} = \sum_{i=1} \| R^* \hat{x}_i^* + t^* - \hat{y}_j \|_2^2.$$  

3) **Registration Supervision**: We utilize the transformation loss $L_{\text{trans}}$ as below to promote the predicted transformation to approach the ground-truth transformation

$$L_{\text{trans}} = \| \hat{R}^\top R^* - I \|_2^2 + \| \hat{t} - t^* \|_2^2.$$  

The final loss function combines the registration-level losses above and the tracking-level losses $L_{\text{track}}$ (including the center-coordinate and rotation-angle regression losses as in [12] and [14])

$$L = L_{\text{cls}} + L_{\text{corr}} + L_{\text{trans}} + L_{\text{track}}.$$  

### IV. Experiments

To verify the effectiveness of our method, we perform extensive experiments on different benchmark datasets and ablation studies in this section. Concretely, we first present our implementation details, the training and testing processing, and the evaluation metric in Section IV-A. Then, we compare our method to some SOTA methods on the KITTI, NuScenes, and Waymo datasets in Sections IV-B–IV-D, respectively. Finally, we present our extensive ablation studies about the proposed modules and the hyperparameter settings in Section IV-E.

#### A. Experimental Settings

1) **Implementation Details**: In our registration module, we set nonlocal iteration times $T$ and feature dimension $d$ to 12 and 128. The semimajor and semiminor axes $a$ and $b$ in (3) are set to 1.6 and 0.4, respectively. We choose 50% points with the highest confidence for inlier selection. In the feature aggregation module, the maximum allowance spatial distance $\sigma$ is set to 0.4. We train the model in 30 epochs, where the first five epochs are used to train the registration network only, and the remaining 25 epochs are used to jointly train the registration and tracking modules. We use the Adam optimizer with a learning rate of 0.001 and weight decay by 0.2 every six epochs for model training. We implement our model with PyTorch and deploy all experiments on a server containing an Intel i5 2.2 GHz CPU and two TITAN RTX GPUs with almost 24 GB per card. For simplicity, we name our Registration-Driven Tracker as RDT.

2) **Training and Testing**: For training, we augment the training samples by adding random offsets on the previous and current ground-truth BBoxes. The former is used to crop the template, and the latter is enlarged by 2 m for search area cropping. For testing, we fuse the points within the first and previous BBoxes to generate the template, while the search area is cropped by the previous BBox enlarged by 2 m. The template and search area points are set to 512 and 1024, respectively.

3) **Evaluation Metrics**: Following [11], we use Success and Precision criteria by one-pass evaluation (OPE) to measure the model performance. Success is defined as the intersection over union (IoU) between the predicted and the ground-truth BBoxes and Precision is the AUC score (area under the curve) for the distance between their centers from 0 to 2 m.

#### B. Evaluation on the KITTI Dataset

We first evaluate our method on the KITTI benchmark [44], a LiDAR-scanned driving scenarios dataset, including 21 outdoor scenes and eight object categories. For a fair comparison, following the processing and data split in [11], we utilize the sequences 0–16 scenes, 17–18 scenes, and 19–20 scenes for training, validation, and testing, respectively. We compare our method to eight SOTA trackers, that is, SC3D [9], P2B [11], 3DSiamRPN [16], BAT [10], PTT [17], PTRR [18], V2B [12].
TABLE I
PERFORMANCE COMPARISONS WITH SOTA TRACKERS ON Car, Pedestrian, Van, AND Cyclist CATEGORIES FROM THE KITTI AND NuScenes DATASETS

| Category | Success |
|----------|---------|
| Frame Num. | Car | Pedestrian | Van | Cyclist | Mean |
| SC3D [9] | 41.3 | 18.2 | 40.4 | 41.5 | 31.2 |
| P2B [11] | 56.2 | 28.7 | 40.8 | 32.1 | 42.4 |
| LTR [43] | 65.0 | 33.2 | 35.8 | 66.2 | 48.7 |
| BAT [10] | 60.5 | 42.1 | 52.4 | 33.7 | 51.2 |
| PTT [17] | 67.8 | 44.9 | 43.6 | 37.2 | 55.1 |
| V2B [12] | 70.5 | 48.3 | 50.1 | 40.8 | 58.4 |
| CAT [19] | 66.6 | 51.6 | 53.1 | 67.0 | 58.9 |
| RDT (ours) | 71.8 | 56.4 | 60.4 | 72.8 | 64.1 |

| Category | Precision |
|----------|-----------|
| Frame Num. | Car | Pedestrian | Van | Cyclist | Mean |
| SC3D [9] | 57.9 | 37.8 | 47.0 | 70.4 | 48.5 |
| P2B [11] | 72.8 | 49.6 | 48.4 | 44.7 | 60.0 |
| LTR [43] | 77.1 | 56.8 | 45.6 | 89.9 | 65.8 |
| BAT [10] | 77.7 | 70.1 | 67.0 | 45.4 | 72.8 |
| PTT [17] | 81.8 | 72.0 | 52.5 | 47.3 | 74.2 |
| V2B [12] | 81.3 | 73.5 | 58.0 | 49.7 | 75.2 |
| CAT [19] | 81.8 | 77.7 | 69.8 | 90.1 | 79.1 |
| RDT (ours) | 83.2 | 84.1 | 69.7 | 93.7 | 82.6 |

NuScenes

| Category | Success |
|----------|---------|
| Frame Num. | Car | Pedestrian | Truck | Bicycle | Mean |
| SC3D [9] | 25.0 | 14.2 | 25.7 | 17.0 | 21.8 |
| P2B [11] | 31.8 | 18.7 | 21.7 | 17.8 | 26.4 |
| BAT [10] | 30.1 | 18.9 | 23.5 | 16.5 | 25.7 |
| V2B [12] | 36.6 | 19.3 | 31.5 | 18.9 | 30.6 |
| RDT (ours) | 37.1 | 20.4 | 33.4 | 18.1 | 31.4 |

| Category | Precision |
|----------|-----------|
| Frame Num. | Car | Pedestrian | Truck | Bicycle | Mean |
| SC3D [9] | 27.1 | 16.2 | 21.9 | 18.2 | 23.1 |
| P2B [11] | 33.6 | 24.7 | 17.8 | 21.9 | 28.7 |
| BAT [10] | 31.3 | 26.0 | 18.5 | 19.2 | 27.8 |
| V2B [12] | 39.2 | 26.6 | 29.3 | 22.0 | 33.9 |
| RDT (ours) | 39.8 | 29.0 | 30.2 | 22.9 | 35.1 |

TABLE II
PERFORMANCE OF DIFFERENT METHODS ON THE WAYMO OPEN DATASET. EACH CATEGORY IS DIVIDED INTO THREE LEVELS OF DIFFICULTY: EASY, MEDIUM, AND HARD. “MEAN” DENOTES THE AVERAGE RESULTS OF THREE DIFFICULTY

| Category | Split |
|----------|-------|
| Frame Num. | Easy | Medium | Hard | Mean |
| SC3D [9] | 67832 | 61252 | 56647 | 185731 |
| P2B [11] | 57.1 | 52.0 | 47.9 | 52.6 |
| BAT [10] | 61.0 | 53.3 | 48.9 | 54.7 |
| V2B [12] | 64.5 | 55.1 | 50.1 | 57.6 |
| RDT (ours) | 65.5 | 58.8 | 52.4 | 58.3 |

| Category | Precision |
|----------|-----------|
| Frame Num. | Easy | Medium | Hard | Mean |
| SC3D [9] | 85280 | 82253 | 74219 | 241752 |
| P2B [11] | 18.1 | 17.8 | 17.7 | 17.9 |
| BAT [10] | 19.3 | 17.8 | 17.2 | 18.2 |
| V2B [12] | 27.9 | 22.5 | 20.1 | 23.7 |
| RDT (ours) | 28.6 | 24.2 | 21.3 | 24.9 |

| Category | Success |
|----------|---------|
| Frame Num. | Easy | Medium | Hard | Mean |
| SC3D [9] | 65.4 | 60.7 | 58.5 | 61.7 |
| P2B [11] | 68.3 | 60.9 | 57.8 | 62.7 |
| BAT [10] | 71.5 | 63.2 | 62.0 | 65.9 |
| V2B [12] | 71.5 | 63.2 | 62.0 | 65.9 |
| RDT (ours) | 72.3 | 64.0 | 62.8 | 66.5 |

| Category | Precision |
|----------|-----------|
| Frame Num. | Easy | Medium | Hard | Mean |
| SC3D [9] | 43.9 | 36.2 | 33.1 | 37.9 |
| P2B [11] | 50.8 | 49.8 | 49.2 | 50.0 |
| BAT [10] | 52.6 | 51.9 | 51.2 | 52.2 |
| V2B [12] | 70.1 | 62.9 | 62.9 | 62.9 |
| RDT (ours) | 71.8 | 83.2 | 83.2 | 83.2 |

TABLE III
ABLATION STUDIES OF DIFFERENT COMPONENTS ON THE Car CATEGORY FROM THE KITTI DATASET. TSNR: TRACKING-SPECIFIC NONLOCAL REGISTRATION; TSNonlocal: SPATIAL DISTANCE-CONSTRAINED NONLOCAL MODULE; Classifier: IQLIER CLASSIFIER; RM: REGISTRATION MATCHING; Sinkhorn: Sinkhorn Optimization

| TSNR | TSNonlocal | Classifier | RM | Sinkhorn | Succ. | Prec. |
|------|------------|------------|----|----------|------|------|
| ✓    | ✓          | ✓          | ✓  |          | 68.7 | 78.7 |
| ✓    | ✓          | ✓          | ✓  |          | 70.3 | 80.9 |
| ✓    | ✓          | ✓          | ✓  |          | 69.7 | 80.2 |
| ✓    | ✓          | ✓          | ✓  |          | 69.4 | 79.7 |
| ✓    | ✓          | ✓          | ✓  |          | 70.1 | 82.9 |
| ✓    | ✓          | ✓          | ✓  |          | 71.8 | 83.2 |

The performance of our method is shown in Table I. Overall, our method can achieve an impressive performance advantage on both Success and Precision criteria compared to other trackers. Notably, compared to V2B that also uses the CenterPoint, such as a detection head like us, our method presents a superior performance on all categories, such as the Pedestrian category (Succ: 48.3→56.4, Prec: 73.5→84.1). This performance gain mainly benefits from our robust feature matching between the template and the search area after registration operation. Also, the proposed RSFA can efficiently transfer the target information into the search area to generate the discriminative target-specific feature for reliable object localization.

C. Evaluation on the NuScenes Dataset

We further test our method on the NuScenes benchmark [45], which is also an outdoor dataset with 1000 driving scenes and 23 annotated object categories. Following [12], we split it into 750 training sequences and 150 validation sequences, where we only evaluate the trained model on the latter since the annotations for its test split are not accessible.
We compare our tracker to three representative SOTA tackers, that is, P2B, BAT, and V2B. All tested trackers, including RDT (ours), utilize the provided training split for model training. As shown in Table I, our method can still outperform all trackers on both Success and Precision criteria in most categories. Notably, different from the 64-beam LiDAR used in the KITTI dataset, NuScenes is just scanned by the 32-beam. Therefore, the existing data gap between the KITTI and the NuScenes largely degrades our generalization precision and RDT can achieve limited performance improvement.

D. Evaluation on the Waymo Dataset

The Waymo dataset [46] is a large-scale outdoor dataset containing 150 scenes for testing, where the number of frames and the scene complexities are significantly beyond the KITTI and NuScenes datasets. To evaluate the generalization ability of our method, we directly generalize the model learned by the KITTI to the Waymo dataset. As shown in Table II, we test on Vehicle and Pedestrian categories, and our method outperforms other methods (P2B, BAT, and V2B) in terms of different difficulty levels (easy, medium, and hard), which demonstrates the strong generalization ability of our model.

E. Ablation Study

1) Registration Module and Its Components: To verify the effectiveness of our TSNR module, we remove it along with its related components containing TSNonlocal, Classifier, and RM. Notably, RM originates from the aligned template X and the search area Y which are achieved by performing the registration module TSNR. Therefore, RM is related to the TSNonlocal module. We test its performance on the representative Car category from the KITTI dataset. As shown in the first row of Table III, without TSNonlocal, the Success and Precision scores of our method degrade significantly (Succ: 71.8→68.7 and Prec: 83.2→78.7). Furthermore, we test the performance gain from TSNonlocal by replacing it with a traditional nonlocal module without spatial distance constraint \( \beta_{i,j} \) in (1). The second row of Table III shows that the used spatial distance constraint can bring 1.5 and 2.3 performance gain on Success and Precision criteria, respectively, which demonstrates that enhancing our nonlocal module by tracking-specific cross-attention regularization is effective. Moreover, we test the inlier classifier (Classifier) in TSNonlocal. The third row of Table III shows that outlier rejection can bring impressive gains for our method since the outliers may largely interfere with the registration and thus degrade the final tracking performance. Finally, we evaluate the registration precision of the TSNR module by registering templates and search areas, which are cropped by the ground-truth BBox on each frame.

Table IV verifies that our registration module can effectively bring significant Success and Precision improvement by robustly aligning the template to the search area.

2) Registration-Aid Sinkhorn Feature Aggregation: The RSFA module focuses on using the Sinkhorn optimization (Sinkhorn) and the registration-aided matching refinement (i.e., RM) to correct the feature matching matrix for robust target-specific feature aggregation. The fourth row of Table III shows that RM can bring almost Succ: 2.4 and Prec: 3.5 performance gain, which verifies that registration-aided spatial distance map facilitates a more reliable feature matching construction for target-specific feature learning. In addition, as shown in the fifth row of Table III, by replacing the traditional Cosine distance-based feature matching, the Sinkhorn optimization-based matching map can also achieve a significant performance improvement, which is mainly due to the robustness to outliers of the Sinkhorn feature matching. Moreover, we investigate more alternative feature backbones, including PointNet [47], PointCNN [48], DGCNN [49], and Hybrid (hybrid feature descriptor used in RPMNet) [42], in our feature aggregation module for a more comprehensive ablation study. Their quantitative results are listed in Table V. It can be observed that benefitting from the effective hierarchical feature aggregation mechanism, PointNet++ can achieve the best tracking performance. Finally, to verify that the feature aggregation can solve the error accumulation during 3-D tracking, we compare the tracking performance of TSNonlocal (only registration) and RDT (TSNonlocal++ RSFA) in Table VI. It can be observed that since registration is incapable of handling error accumulation, only the TSNonlocal-based tracker would be restricted to a very limited tracking precision. Conversely, RDT, which combines 3-D registration and feature aggregation-based localization modules, achieves significant performance improvement. These results quantitatively validate that feature aggregation-based object localization can solve the problem of error accumulation.

3) Parameter Setting: We further evaluate the performance changes under different settings of semimajor axis \( a \) of TSNonlocal (3) and maximum allowance distance \( \sigma \) of RM in Fig. 4 (7). It is noted that we directly generalize the pretrained model to these parameters without retraining. Com-
pared to the SOTA V2B, our method outperforms it in most settings without specific training, which demonstrates the robustness of our model on parameter selection. Also, in Table VII, we test the Success/Precision scores under varying ratios of hypothetical inliers (with top-k highest inlier probabilities). It can be observed that compared to our default setting (50% inlier ratio), the 62.5% inlier ratio can achieve the highest tracking scores. In addition, we find that too small or too large inlier ratios would lead to bad tracking performance. The main reasons are: 1) too small inlier ratios mean that fewer points will be chosen from the template and the search area, leading to low-overlapped point cloud pair and thus increasing the registration difficulty and 2) too large inlier ratios would introduce more outliers into registration, thereby degrading the registration precision.

4) Qualitative Comparison and Registration Visualization: We show some representative visual comparisons with V2B on the Car category of KITTI in Fig. 5, where the top and bottom sequences show the dense-to-sparse and sparse-to-dense sequences, respectively. It can be observed that our method can handle well in the challenging scenes with sparse LiDAR scanning. Due to our tracking-specific registration design, our method can precisely align the template to the search area, whether in dense scenes (columns 1–3) or sparse scenes (columns 4–6).

5) Performance Evaluation in Sparse Scenes: To further evaluate our tracking robustness in sparse scenes, we report the performance comparison at different point intervals of the KITTI dataset in Table VIII. It can be observed that our model can still significantly outperform other SOTA trackers under different sparse levels (except for point interval [0, 150) in the Car category). Particularly, compared to V2B, our model can even achieve 7% and 8.5% performance gain on the averaged Success and Precision scores, respectively.

6) Inference Speed: We use FPS for inference speed evaluation. For a fair comparison, we run each tracker on the Car class from KITTI with the same configuration (a server with a TITAN RTX GPU). The tracking speed of our method is 13 FPS, and SC3D, P2B, BAT, 3DSiamRPN, PTT, PTTR, and V2B can achieve 3 FPS, 28 FPS, 30 FPS, 12 FPS, 24 FPS, 31 FPS, and 22 FPS, respectively. Although our tracking speed tends to be slower than P2B, BAT, PTT, PTTR, and V2B due to the time cost of the registration, the registration module can bring a significant performance gain at a tolerable speed.

V. CONCLUSION

In this article, we proposed a novel point cloud registration-driven robust feature-matching framework for 3-D single object Siamese tracking. Our framework includes two key components: the TSNR module and the RSFA module. The former aims to construct reliable feature matching by spatially aligning the template to the search area, whereas the spatial distance constraint of the corresponding points is integrated into its nonlocal module for robust feature learning. The latter combines the Sinkhorn optimization and the registration-aided spatial distance map to establish a robust feature-matching framework.
map. Finally, the constructed feature-matching map guides the transfer of the target information into the search area for object localization. Extensive experiments on the benchmark datasets demonstrate the effectiveness of our proposed method.

REFERENCES

[1] W. Luo, B. Yang, and R. Urtasun, “Fast and furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 3569–3577.

[2] A. I. Comport, E. Marchand, and F. Chaumette, “Robust model-based tracking for robot vision,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Sep. 2004, pp. 692–697.

[3] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, “PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 5588–5597.

[4] T. Wang et al., “Spatio-temporal point process for multiple object tracking,” IEEE Trans. Neural Netw. Learn. Syst., vol. 34, no. 4, pp. 1777–1788, Apr. 2023.

[5] L. Jiao, D. Wang, Y. Bai, P. Chen, and F. Liu, “Deep learning in visual learning: A review,” IEEE Trans. Neural Netw. Learn. Syst., vol. 34, no. 9, pp. 5497–5516, Sep. 2021.

[6] J. Ma, A. Fan, X. Jiang, and G. Xiao, “Feature matching via motion-consistency driven probabilistic graphical model,” Int. J. Comput. Vis., vol. 130, no. 9, pp. 2249–2264, Sep. 2022.

[7] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, “Image matching from handicrafted to deep features: A survey,” Int. J. Comput. Vis., vol. 127, no. 5, pp. 512–531, May 2019.

[8] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, “Locality preserving matching,” Int. J. Comput. Vis., vol. 129, no. 5, pp. 512–531, May 2020.

[9] S. Giancola, J. Zarzar, and B. Ghanem, “Leveraging shape completion for 3D Siamese tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 1359–1368.

[10] C. Zhang et al., “Box-aware feature enhancement for single object tracking on point clouds,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 13179–13188.

[11] H. Qi, C. Feng, Z. Cao, F. Zhao, and Y. Xiao, “P2B: Point-to-box network for 3D object tracking in point clouds,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 6328–6337.

[12] L. Hui, L. Wang, M. Cheng, J. Xie, and J. Yang, “3D Siamese voxel-to-bev tracker for sparse point clouds,” in Proc. NeurIPS, vol. 34, 2021.

[13] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” 2017, arXiv:1706.02413.

[14] T. Yin, X. Zhou, and P. Krähenbühl, “Center-based 3D object detection and tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 11779–11788.

[15] C. R. Qi, O. Litany, K. He, and L. Guibas, “Deep Hough voting for 3D object detection in point clouds,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9276–9285.

[16] Z. Fang, S. Zhou, Y. Cui, and S. Scherer, “3D-SiamRPN: An end-to-end learning method for real-time 3D single object tracking using raw point cloud,” IEEE Sensors J., vol. 21, no. 4, pp. 4995–5011, Feb. 2021.

[17] J. Shan, S. Zhou, Z. Fang, and Y. Cui, “PTT: Point-track-transformer module for 3D single object tracking in point clouds,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Sep. 2021, pp. 1310–1316.

[18] A. Zhou et al., “PTTR: Relational 3D point cloud object tracking with transformer,” 2021, arXiv:2112.02857.

[19] J. Gao, X. Yan, W. Zhao, Z. Lyu, Y. Liao, and C. Zheng, “Spatio-temporal contextual learning for single object tracking on point clouds,” IEEE Trans. Neural Netw. Learn. Syst., early access, Jun. 6, 2023, doi: 10.1109/TNNLS.2022.323562.

[20] H. Wu, W. Han, C. Wu, X. Li, and C. Wang, “3D multi-object tracking in point clouds based on prediction confidence-guided data association,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 6, pp. 5668–5677, Jun. 2021.

[21] T. Wang et al., “RJMOT: A real-time 3D multi-object tracker and a new large-scale dataset,” in Proc. IEEE/CVF Int. Conf. Intell. Robots Syst. (IROS), Oct. 2020, pp. 10335–10342.

[22] A. Kim, A. Olep, and L. Leal-Taixé, “EagerMOT: 3D multi-object tracking via sensor fusion,” 2021, arXiv:2104.14682.

[23] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, “From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 8, pp. 2647–2664, Aug. 2021.

[24] S. Shi, X. Wang, and H. Li, “PointRCNN: 3D object proposal generation and detection from point cloud,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019.

[25] X. Weng, J. Wang, D. Held, and K. Kitani, “3D multi-object tracking: A baseline and new evaluation metrics,” 2019, arXiv:1907.03961.

[26] Y. Wang, K. Kitani, and X. Weng, “Joint object detection and multi-object tracking with graph neural networks,” 2020, arXiv:2006.13164.

[27] A. Patil, S. Malla, H. Gang, and Y.-T. Chen, “The H3D dataset for full-surround 3D multi-object detection and tracking in crowded urban scenes,” in Proc. Int. Conf. Robot. Autom. (ICRA), May 2019, pp. 9552–9557.

[28] P. J. Besl and N. D. McKay, “Method for registration of 3-D shapes,” Proc. SPIE, vol. 1611, pp. 586–606, 1992.

[29] J. Yang, H. Li, and Y. Jia, “Go-ICP: Solving 3D registration efficiently and globally optimally,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013, pp. 1457–1464.

[30] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, “The trimmed iterative closest point algorithm,” in Object Recognition Supported by User Interaction for Service Robots, 2002.

[31] Y. Wang and J. Solomon, “Deep closest point: Learning representations for point cloud registration,” in Proc. IEEE/CVF Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3522–3531.

[32] Y. Wang and J. M. Solomon, “PRNet: Self-supervised learning for partial-to-partial registration,” 2019, arXiv:1910.12240.

[33] J. Li, C. Zhang, Z. Xu, H. Zhou, and C. Zhang, “Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration,” in Proc. ECCV, 2020.
[34] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, “PointNetLK: Robust & efficient point cloud registration using PointNet,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 7156–7165.

[35] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz, “DeepGMR: Learning latent Gaussian mixture models for registration,” in Proc. ECCV, 2020.

[36] T. Wan et al., “RGB-D point cloud registration based on salient object detection,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 8, pp. 3547–3559, Aug. 2022.

[37] H. Jiang, Y. Shen, J. Xie, J. Li, J. Qian, and J. Yang, “Sampling network guided cross-entropy method for unsupervised point cloud registration,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 6108–6117.

[38] H. Jiang, J. Xie, J. Qian, and J. Yang, “Planning with learned dynamic model for unsupervised point cloud registration,” 2021, arXiv:2108.02613.

[39] D. Bauer, T. Patterson, and M. Vincze, “ReAgent: Point cloud registration using imitation and reinforcement learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 11821–11830.

[40] Y. Cui, Z. Fang, J. Shan, Z. Gu, and S. Zhou, “3D object tracking with transformer,” 2021, arXiv:2110.14921.

[41] P. Sun et al., “Scalability in perception for autonomous driving: Waymo open dataset,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 6118–6128.

[42] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, “PointNet+Deep learning on point sets for 3D classification and segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 77–85.

[43] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, “PointCNN: Convolution on x-transformed points,” in Proc. NIPS, 2018.

[44] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic graph CNN for learning on point clouds,” ACM Trans. Graph., vol. 38, no. 5, pp. 1–12, Oct. 2019.

**Kaihao Lan** received the B.S. degree in computer science and technology from Hangzhou Dianzi University, Hangzhou, China, in 2020, and the M.S. degree in computer technology from Nanjing University of Science and Technology, Nanjing, China, in 2023. His current research interests include 3-D computer vision and autonomous driving.

**Haobo Jiang** received the Ph.D. degree in control science and engineering from the School of Computer Science and Engineering, Nanjing University of Science and Technology (NUST), Nanjing, China. His current research interests focus on reinforcement learning and 3-D computer vision, including robotic control and planning, 3-D point cloud registration, and 3-D object tracking.

**Le Hui** received the Ph.D. degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology (NUST), Nanjing, China, in 2022. He is currently an Assistant Professor at Northwestern Polytechnical University, Xi’an, China. His current research interests include 3-D point cloud processing, such as semantic segmentation, single object tracking, place recognition, and point cloud generation.

**Guangyu Li** received the B.S. degree from the China University of Mining and Technology, Xuzhou, China, in 2008, the M.S. degree from Tongji University, Shanghai, China, in 2011, and the Ph.D. degree in computer science from the University of Paris-Sud, Paris, France, in 2015. He is currently working as an Associate Professor at Nankai University, Tianjin, China. His current research interests include wireless networks and computer vision.

**Jin Xie** received the Ph.D. degree from the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. He is currently a Professor at the Nanjing University of Science and Technology (NUST), Nanjing, China. Before joining NUST, he was a Research Scientist at New York University Abu Dhabi, Abu Dhabi, United Arab Emirates. He is focusing on 3-D computer vision with deep-learning methods. He has authored articles in top conferences and journals, including CVPR, ICCV, ECCV, NeurIPS, and IEEE Transactions on Pattern Analysis and Machine Intelligence. His current research interests include computer vision, machine learning, and robotics.

Dr. Xie has served as a PC/SPC Member for CVPR, ICCV, ECCV, AAAI, and ICAL. He was a Special Issue Chair of ACPR 2017 and a Guest Editor of Pattern Recognition.

**Shangbing Gao** received the Ph.D. degree in computer science and technology from the Nanjing University of Science and Technology, Nanjing, China, in 2014. He is currently a Professor with the Faculty of Computer and Software, Huaiyin Institute of Technology, Huai’an, China. His current research interests include pattern recognition and computer vision.

**Jian Yang** received the Ph.D. degree from Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002, on the subject of pattern recognition and intelligence systems.

In 2003, he was a Post-Doctoral Researcher at the University of Zaragoza, Zaragoza, Spain. From 2004 to 2006, he was a Post-Doctoral Fellow at the Biometrics Center, Hong Kong Polytechnic University, Hong Kong. From 2006 to 2007, he was a Post-Doctoral Fellow at the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA. Now, he is a Chang-Hiang Professor at the School of Computer Science and Technology, NUST. He has authored more than 200 scientific papers in pattern recognition and computer vision. His papers have been cited over 30 000× in the Scholar Google. His current research interests include pattern recognition, computer vision, and machine learning.

Dr. Yang is a fellow of IAPR. He is an Associate Editor of Pattern Recognition, Pattern Recognition Letters, IEEE Transactions on Neural Networks and Learning Systems, and Neurocomputing.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.