International Journal of Computer Vision (2026) 134:333
https://doi.org/10.1007/s11263-026-02921-0

End-to-End LiDAR-Camera Calibration via Multi-Modal Correspondences Estimation and Explicit BEV Alignment

Lorenzo Cipelli[1] · Filippo D'Addeo[2] · Emanuele Ghelfi[3] · Marcello Ceresini[1] · Andrea Bertogalli[1] · Federico Pirazzoli[1] · Massimo Bertozzi[1]

Received: 7 January 2026 / Accepted: 11 June 2026

Abstract

In this work, we present a Bird's Eye View (BEV) Alignment approach for the LiDAR-Camera calibration task. Building upon previous BEV-based work, we extract sensor-wise BEV features from each input modality using domain-specific architectures. Then, we employ a CNN-based encoder to align the two BEVs and estimate the calibration matrix. However, corresponding 2D and 3D features may be spatially distant in BEV space, and as a consequence the encoder alone might struggle to learn the height dimension and estimate the correct registration matrix. To address this, we introduce an implicit alignment step to cross-attend the downsampled 3D features with those from RGB for computing point-to-pixel correspondences and estimating a coarse calibration matrix. To improve the implicit alignment, we also enforce the prediction of correct point-to-pixel correspondences by direct supervision of the similarity matrix computed into the cross attention module. Then, the coarsely aligned 3D features and the RGB features are fed to the BEV Alignment step, in which the CNN-based encoder refines the coarse estimate into a final, more accurate calibration matrix. Notably, both the steps are optimized in an end-to-end fashion. Our method significantly outperforms previous point-to-pixel matching methods, achieving state-of-the-art calibration accuracy. On the KITTI and nuScenes benchmarks, our method reduces the Relative Rotation Error (RRE) by 74% and 79%, and the Relative Translation Error (RTE) by 90% and 95%, respectively, compared to previous methods.

Keywords Target-less lidar-camera calibration · Learning-based calibration · Image-to-point cloud registration · Transformer-based calibration · Sensor fusion

1 Introduction

Image-to-Point Cloud registration (Fig. 1) is crucial for many application in Autonomous Driving, 3D Computer Vision, Augmented/Virtual Reality, and robotics domains. The goal is to estimate the rigid transformation, in terms of rotation and translation, that precisely aligns the projection of a LiDAR point cloud with a reference RGB camera image. Early works (Li & He Lee, 2021; Ren et al., 2023) focus on extracting 2D and 3D features from RGB images and point clouds to establish 2D-3D correspondences. However, these methods often struggle with matching image and point cloud features, as different modalities and architectures do not inherently

Communicated by Francesco Ragusa.

L. Cipelli, F. D'Addeo: These authors contributed equally to this work.

✉ Filippo D'Addeo
filippo.daddeo2@unibo.it

Lorenzo Cipelli
lorenzo.cipelli@unipr.it

Emanuele Ghelfi
eghelfi@ambarella.com

Marcello Ceresini
marcello.ceresini@unipr.it

Andrea Bertogalli
andrea.bertogalli@unipr.it

Federico Pirazzoli
federico.pirazzoli@unipr.it

Massimo Bertozzi
massimo.bertozzi@unipr.it

[1] Department of Engineering and Architecture, University of Parma, Parma, Italy

[2] Department of Industrial Engineering, University of Bologna, Bologna, Italy

[3] VisLab srl, an Ambarella Inc. company, Parma, Italy

Published online: 03 July 2026

produce a shared feature space for seamless feature matching. To address this challenge, several attempts have been made: Zhou et al. (2023) introduce a new voxel-based branch to learn related representations, Bie et al. (2025) use virtual point cloud to match corresponding representations of points, Li et al. (2025) introduce correspondence queries to extract keypoints, and Yuan et al. (2025) leverage a BEV space to fuse the features from both the input modalities.

In this work, inspired by the detection literature (Philion & Fidler, 2020; Harley et al., 2023; Li et al., 2024; Yang et al., 2023; Lang et al., 2019; Yin et al., 2021), we choose to follow the idea proposed by Yuan et al. (2025) to employ Bird's Eye View (BEV) space for fusing the RGB and LiDAR features. As matter of fact, we aim to generate two BEV feature representations of the same scene, one from a sparse LiDAR point cloud and the other from an RGB image. However, in contrast with (Yuan et al., 2025), we further choose to formulate the Image-to-Point Cloud task as a *Bird's Eye View alignment* task, rather than leverage BEV features only for fusing the two input modalities, since the two BEV representations capture semantically related scene structures even if they are derived from different modalities. This way, we can accommodate rotations up to $\pm 360^\circ$ as required by the standard point-based LiDAR-Camera calibration benchmark (Li & He Lee, 2021; Ren et al., 2023; Zhou et al., 2023; Bie et al., 2025; Li et al., 2025).

The BEV alignment task is achieved by fusing the height dimension of the two BEVs together with the channels dimension of the two top views and feed them to a convolutional encoder, which is trained together with a rotation and translation heads to predict the rigid transformation that aligns the LiDAR point cloud with the RGB image. Notably, at this stage, we do not explicitly align the two BEVs. Instead, we let the convolutional encoder implicitly learn the calibration parameters from the mis-aligned BEV feature maps by supervising the whole network only with the ground truth rigid transformations. We argue that this is feasible due to the large receptive field of CNNs, which enables a global understanding of the BEV feature maps and allows the model to reason about the scene geometry for an accurate calibration.

Although this strategy seems to be already effective as proven by our experiments in Section 4.4, a key improvement can be done by reasoning on non-standard benchmarks. Indeed, due to any given initial mis-registration, corresponding image and point cloud features may be spatially distant in BEV space, and the BEV Alignment alone might struggle to learn the height dimension and estimate the correct registration matrix. For this reason, we propose a novel Implicit Alignment module to estimate an intermediate registration matrix to leverage the BEV formulation and coarsely align the BEV features of the LiDAR with those from the RGB, allowing the CNN-based encoder to refine the predicted calibration matrix. The implicit alignment is achieved by employing a transformer decoder module to directly cross-attend the RGB features with the downsampled LiDAR features for computing the point-to-pixel correspondences, and feed them to a single-query cross attention-based prediction head, responsible for the coarse calibration matrix estimate. Afterward, inspired by recent multi-modal alignment techniques (Radford et al., 2021), we adopt a specific loss to guide the cross attention mechanism to output the correct pixel-to-point correspondences, stabilizing the training and facilitating the prediction head task. As a result of this intermediate alignment, corresponding cells between the two BEV feature maps become spatially closer in the fusion space, simplifying the BEV Alignment task for the encoder and improving the overall registration performance. Notably, in contrast with previous two-step algorithm (Li et al., 2025), we train both the Implicit Alignment and the BEV Alignment modules in an end-to-end fashion.

To summarize, out contributions are:

- We propose a novel framework built upon detection literature and latest advances in Image-to-Point Cloud registration task that frames the Camera-LiDAR calibration task as a BEV alignment problem.
- Leveraging the BEV formulation, we propose a two-step alignment algorithm. Firstly, an Implicit Alignment performs an attention-based alignment by learning correct point-pixel correspondences to estimate a coarse calibration, which is then exploited in the BEV alignment step to predict the final and refined calibration parameters. Notably, both steps are trained in an end-to-end fashion.
- Our model achieves state-of-the-art performance in both KITTI and nuScenes datasets, surpassing previous methods while being more robust.
- Our model achieves state-of-the-art performance also in the 6 Degrees of Freedom benchmark, surpassing previous methods and other projection-based approaches.

2 Related Work

2.1 Image-to-Point Cloud Registration

The problem of LiDAR-Camera extrinsic calibration has been widely explored, with early solutions primarily based on target-based and target-less methods. Target-based approaches (Cheng et al., 2023; Sugimoto et al., 2004; Wang et al., 2011; Bertogalli et al., 2025) rely on predefined calibration objects and optimization techniques to align sensor data. In contrast, target-less methods (Levinson & Thrun, 2013; Pandey et al., 2012; Yuan et al., 2021a) leverage environmental features or statistical models, offering greater flexibility and resulting also suitable for online applications. Our method belongs to this latter category. Recent state-of-the-art

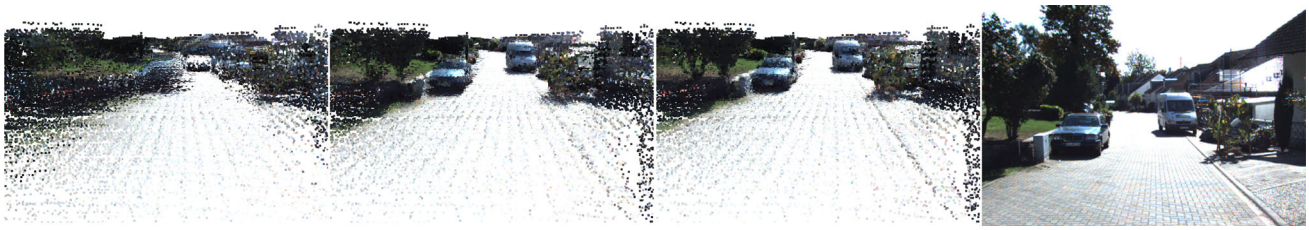


Fig. 1 Image-to-Point registration overview. From left to right: initial mis-registration, final registration, original (ground truth) registration, corresponding frame. For illustrative purposes the point clouds position are the same across all images, the mis-registration is applied to the RGB color sampling from the corresponding image

target-less approaches, particularly those based on correspondences (Yuan et al., 2021b; Koide et al., 2023; Huang et al., 2024), have demonstrated impressive registration accuracy. However, they often require extensive pre-processing, accumulation of multiple frames, and meticulous parameter tuning for each scene, making them impractical for dynamic driving environments where scenes change rapidly and fast inference is essential. In contrast, our approach eliminates the need for such pre-processing or frame accumulation, enabling faster and more adaptable calibration.

2.2 Projection-based Methods

The projection-based methods estimate the sensor calibration by projecting the LiDAR point cloud onto the image plane, generating a mis-registered sparse depth map, and training a neural network to regress the calibration parameters. RegNet (Schneider et al., 2017) employs two separate 2D backbones to extract features from the RGB image and the sparse depth map, which are then fused for calibration estimation. Similarly, CalibNet (Iyer et al., 2018) enforces geometric consistency by ensuring that the projected sparse depth map, transformed with predicted extrinsics, aligns with the ground-truth sparse depth. Building on these foundations, later methods introduced more advanced architectures: LCCNet (Lv et al., 2021) incorporates a feature matching layer to correlate RGB and depth features, while CalibFormer (Xiao et al., 2024) leverages attention mechanisms for feature fusion. Despite their effectiveness, projection-based methods have a key limitation: when there are no correspondences between the sparse depth projection and RGB, registration becomes impossible. In contrast, our approach directly processes RGB images and point clouds, accommodating uncalibrated sensors with rotations up to $\pm 360°$.

2.3 Point-based Methods

Unlike projection-based methods, point-based algorithms operate directly in the 3D domain, processing point clouds instead of 2D sparse depth maps. DeepI2P (Li & He Lee, 2021) is one of the first methods to leverage 3D architectures (Li et al., 2018; Qi et al., 2017b) to directly extract features from the point cloud. In its first stage, DeepI2P classifies whether each point lies within the camera frustum, or not. The second stage addresses the inverse camera projection problem by coarsely classifying points into corresponding image patches. Finally, a RANSAC-based PnP algorithm is applied to the grid classification output. Similarly, CorrI2P (Ren et al., 2023) proposes an early classification stage to detect the overlapping region between the RGB and the point cloud while introducing cross-attention into the framework. In the second stage, the camera pose can be obtained by applying EPnP (Lepetit et al., 2009) within RANSAC on the dense correspondences between 2D and 3D features. These methods often struggle with matching image and point cloud features, as different modalities and architectures (CNNs for images and MLPs for point clouds) do not inherently produce a shared feature space for seamless feature matching. To address this challenge, VP2P-Match (Zhou et al., 2023) introduces voxel-based representations into the framework exploiting an additional backbone. The voxel branch captures spatially local patterns, akin to the image branch, ensuring that pixels and voxels share similar feature structures. Finally, a differentiable probabilistic PnP solver (Chen et al., 2022) estimates the camera pose, making the network fully end-to-end differentiable. Later, CurrI2P (Lin et al., 2025) improves both VP2P-Match and CorrI2P through the curriculum learning technique. However, these methods tend to be slow due to the iterative nature of the outlier filtering methods such as RANSAC and the use of differentiable solvers, while also heavily relying on the classification of which points fall into the camera frustum. For this reason, GraphI2P (Bie et al., 2025) focuses on the feature consistency by estimating the camera pose given the LiDAR point cloud and a virtual cloud, obtained via depth estimation (Bhat et al., 2023), and by leveraging a point sampling on an unified spherical representation to build consistent features patterns, facilitating both the feature fusion and the graph-based correspondence selection phase. Finally, ICLM (Li et al., 2025) introduces a set of learnable correspondence queries specialized as 2D and

3D keypoints detectors for estimating the final pose. Similarly to previous approaches, we operate in a setting where features from the 2D and 3D domains are extracted from two independent networks and we adopt the same benchmark of point-based methods, consisting of a random vertical axis rotation in the range $\pm 360°$ and a random forward-backward and left-right axis translation in the range ± 10m.

2.4 BEV-based Methods

Unlike point-based approaches, BEV-based algorithms operate in the Bird's Eye View (BEV) fusion space, rather than directly operating in the 3D domain, in order to help mitigating the domain gap between the 3D point cloud and the 2D images. BEVCalib (Yuan et al., 2025) projects both the RGB image and the mis-registered LiDAR point cloud into a Bird's Eye View (BEV) space by lifting and splatting the correspondent features (Philion & Fidler, 2020). The resulting BEV features are then employed for estimating the final calibration matrix. While this representation introduces a novel perspective for the task, BEVCalib still inherits a key limitation common to projection-based methods: a minimal overlap between the projected RGB and LiDAR features is still required to make the registration possible. To address this limitation, our method introduces two complementary modules: an Implicit Alignment module, which brings the two modalities closer together, and a BEV Alignment module, which refines any remaining misalignment. This dual-stage alignment allows our approach to handle the transformations defined by the standard benchmark, consisting of a random vertical axis rotation in the range $\pm 360°$ and a random forward-backward and left-right axis translation in the range ± 10m.

3 Method

3.1 Background

As illustrated in Fig. 2, our network takes as input both an image $I \in \mathbb{R}^{H \times W \times 3}$ and a point cloud $P \in \mathbb{R}^{N \times 3}$ collected by an RGB camera and a LiDAR, respectively, where H and W are the height and the width of the image and N is the number of points in the cloud. Following previous works (Harley et al., 2023; Zhou et al., 2023; Li et al., 2025), we adopt two state-of-the-art 2D and 3D branches to independently encode the input modalities, opting for this approach over more complex multi-modal fusion architectures to achieve a lighter network. Building upon the detection literature, we first encode the images with a ResNet-50 (He et al., 2016) 2D backbone and then we employ an FPN (Lin et al., 2017) module to upsample the features C5 of dimensions $H/32 \times W/32 \times C_5$, to the final P4 features F_{rgb} of size $H/16 \times W/16 \times C_{rgb}$. In parallel, we encode the point cloud using a point-based 3D backbone, akin PointNet++ (Qi et al., 2017b), which consists of a downsampling and an upsampling modules. Specifically, the downsampling applies a sequence of set abstraction layers, each of them composed by a sampling and grouping mechanism to reduce the cloud dimension and compute the features $F'_{pts} \in \mathbb{R}^{N_d \times C_d}$, while the upsampling employs a set of feature propagation modules computing the 3D point-wise features $F_{pts} \in \mathbb{R}^{N \times C}$. We refer to the appendix material for a detailed graphical representation of both the backbones. Subsequently, to ensure the same features dimensionality C for the RGB and both the point features, we encode the F_{rgb} with a single convolutional layer and both F'_{pts} and F_{pts} with a single linear layer. Then, after lifting and projecting both the 2D RGB and the 3D point cloud features into two 3D Bird's Eye View (BEV) spaces, we introduce a BEV Alignment module (see Section 3.3) to estimate the calibration matrix directly from the mis-registrated BEVs features.

However, due to any given initial mis-registration, corresponding image and point cloud features may be spatially distant in BEV space, and the BEV Alignment alone might struggle to estimate the correct registration matrix. To address this, we propose an intermediate Implicit Alignment module that predicts a coarse transformation matrix to explicitly align the BEV features of the 3D branch with those from the RGB. This alignment occurs before feeding the BEV features into the BEV Alignment module, which then refines the predicted calibration matrix (see Section 3.2).

3.2 Implicit Alignment

Peculiar to our BEV formulation, we can exploit an intermediate transformation matrix to coarsely align the BEVs from the two modalities, in order to facilitate the prediction heads task and better support the BEV height estimation when non-standard transformation along each axis are involved. To this end, inspired by Li et al. (2025), we choose to leverage the transformer decoder (Vaswani et al., 2017) architecture for estimating point-to-pixel correspondences and predicting a coarse registration matrix. Unlike Li et al. (2025), we avoid to randomly initialize any learnable correspondence query. Instead, we directly cross attend the 2D features with the downsampled 3D features F'_{pts}, employing the 3D features as query set to ensure consistency with the softmax operator in the cross attention. This design choice is motivated by the projection relationships: each 3D point corresponds to a single image pixel, whereas each image pixel may correspond to multiple 3D points due to the reduced scale of the image and occlusion effects. Consequently, using 2D features as queries would introduce inconsistencies in the cross attention similarity matrix.

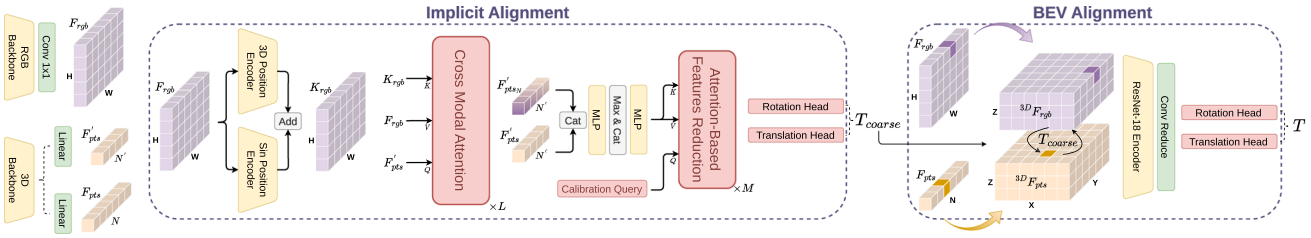


Fig. 2 Overview of our framework. We process RGB images and point clouds independently, and we adopt a cross modal attention-based module to estimate pixel-point correspondences for regressing an initial coarse calibration matrix T_{coarse} (**Implicit Alignment**). In the **BEV Alignment** step, we apply the coarse calibration matrix to the LiDAR coordinates to coarsely align the 3D features to the ones from the RGB. The features from both the modalities are then lifted into two BEV representations, from which a convolutional encoder and prediction heads produce the final calibration matrix T

To facilitate the modality fusion, we follow the common detection tasks practice (Li et al., 2024; Liu et al., 2022) to apply a positional encoding to the RGB features. Specifically, we employ both a 3D Position Encoder (Liu et al., 2022) and a Sinusoidal Position Encoder (Vaswani et al., 2017) to enrich the 2D features with 3D spatial information. To achieve this, we define a grid of 2D image coordinates with dimensions $2 \times H/16 \times W/16$. For each image pixel $[u, v]$, we sample the depth information D using the Linear-Increasing Discretization (LID) (Reading et al., 2021), building the set of coordinates $I \in \mathbb{R}^{3 \times H/16 \times W/16 \times D}$. These are then projected into the camera EDN frame $\{c\}$ through the known inverse intrinsics matrix K^{-1}:

$$\left[{}^{c}x\ {}^{c}y\ {}^{c}z\right]^{T} = K^{-1} \cdot i, \quad \forall i = \left[u\ v\ d\right]^{T} \in I, \tag{1}$$

building the set of 3D camera coordinates ${}^{c}I$. Afterward, we flatten the first and last axes of ${}^{c}I$ to obtain the dimensions of $H/16 \times W/16 \times (3 \cdot D)$, and we feed them to two 1×1 convolutional layers with ReLU non-linearity to get the 3D positional encoding $PE_{3D} \in \mathbb{R}^{H/16 \times W/16 \times C}$. Finally, following Vaswani et al. (2017), also a sinusoidal encoding PE_{sin} is computed and the features $F^{'}_{rgb} \in \mathbb{R}^{H/16 \times W/16 \times C}$ are derived from F_{rgb} as follows:

$$F^{'}_{rgb} = F_{rgb} + PE_{sin} + PE_{3D}. \tag{2}$$

Inspired by Darcet et al. (2023), we introduce an additional learnable key-value token, akin registry token. Indeed, in the cross attention mechanism, we aim to establish correspondences between the queries (represented by 3D features) and the keys (represented by 2D features). However, computing similarity scores between image pixels and 3D points that lie outside the ground truth camera field of view would lead to inaccurate associations and introduce training instability. To mitigate this, we propose a dedicated learnable key-value token $K_{reg} \in \mathbb{R}^{1 \times 1 \times C}$ designed to match all the aforementioned out-of-view points added to the features $F^{'}_{rgb}$ as:

$$KV_{rgb} = F^{'}_{rgb} \oplus K_{reg}, \tag{3}$$

where $\oplus$ corresponds to the concatenation operation.

A transformer decoder T_1 is then employed to effectively establish correspondences between 3D point features and 2D image pixels features. In this setup, the features KV_{rgb} serves both as keys and values while the downsampled 3D features $F^{'}_{pts}$ as queries. Consequently, for each layer $l \in [1, L]$, the query embeddings are refined by attending to the image features that yield the highest similarity score:

$$\begin{aligned} K &= \eta(KV_{rgb}), \\ V &= \mu(KV_{rgb}), \\ Q_0 &= \tau(F^{'}_{pts}), \\ Q_l &= \sigma\left(\frac{Q_{l-1} \cdot K^{T}}{d_{model}}\right) \cdot V, \end{aligned} \tag{4}$$

where η, μ, and τ are linear projections, while σ corresponds to the softmax operation. The final embeddings Q_L are then fused with the input 3D cloud features $F^{'}_{pts}$ in order to explicitly encode the correlation between the predicted point-image correspondences. Afterwards, we employ two successive MLPs modules interleaved with max-pooling operation to progressively fuse and reduce the features. To this end, the features $F^{\prime}_{pts}$ are fed as input to a first MLP module followed by a max-pooling operation. The resulting global features are then concatenated back with the original MLP output to enhance their representation, before being fed as input to a second MLP as follows:

$$\begin{aligned} F^{\prime\prime}_{pts} &= \psi\left(F^{\prime}_{pts} \oplus Q_L\right), \\ F^{\prime\prime\prime}_{pts} &= \phi\left(F^{\prime\prime}_{pts} \oplus \max\left(F^{\prime\prime}_{pts}\right)\right), \end{aligned} \tag{5}$$

with ψ and ϕ being two-layers MLPs, and $\oplus$ the concatenation operation.

Finally, building on the approach from Wang et al. (2025), we adopt a lightweight single-query transformer decoder module T_2 to aggregate the features $F'''_{pts} \in \mathbb{R}^{N_d \times 4C}$, replacing the average-pooling strategy adopted by Li et al. (2025), Yuan et al. (2025). Specifically, the updated 3D features are directly cross-attended with a single learnable query $Q_f \in \mathbb{R}^{1\times 4C}$, designed to assign varying levels of relevance to each feature, thereby selectively emphasizing the most informative ones. Finally, the updated query embeddings of dimensions $1 \times 4C$ serves as input to the translation and rotation prediction heads. Both heads are implemented as two-layers MLPs with ReLU non-linearity. For the translation component, we directly predict the translation vector $t = \left[x\ y\ z\right]^T$. Conversely, for the rotation component, we estimate the sine and the cosine values of the rotation angles along each axis. From these predictions, we construct the corresponding rotation matrices $R_z(\psi)$, $R_y(\phi)$, and $R_x(\theta)$, which are then combined to decode the coarse calibration matrix as:

$$T_{coarse} = \begin{bmatrix} R_z(\psi) \cdot R_y(\phi) \cdot R_x(\theta) & t \\ 0 & 1 \end{bmatrix}, \tag{6}$$

where ψ, ϕ, and θ correspond to yaw, pitch and roll, respectively.

Finally, the transformation matrix T_{coarse} is used to coarsely align the mis-registered LiDAR coordinates $P \in \mathbb{R}^{3\times N}$ before projecting the features $F_{pts} \in \mathbb{R}^{N\times C}$ into the BEV space as described in Section 3.3:

$$p = T_{coarse} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \forall p \in P. \tag{7}$$

As previously discussed, in this module, we aim to predict a coarse registration matrix by estimating the correct point-to-pixel correspondences via direct cross attention between 2D and 3D features. To this end, inspired by Radford et al. (2021), we adopt a specific loss to directly supervise the similarity matrix computed by the transformer decoder T_1 (see Eq. (4)) to encourage the cross attention mechanism to output correct point-pixel correspondences. Such alignment stabilizes the transformer training while facilitating the prediction heads task, allowing them to focus solely on a spatial alignment rather than compensating for mismatched correspondences. During training, we construct the target one-hot 2D-3D correspondences matrix $\hat{S} \in \mathbb{R}^{N_d \times (H/16+1 \times W/16+1)}$ through the known ground truth registration matrix $\left[\hat{R}\ \hat{t}\right]$. This matrix enable us to correctly transform the point cloud into camera coordinates and project them into the image plane to find the corresponding pixel-point pairs:

$$\left[u\ v\ d\right]^T = K \cdot \left(\hat{R} \cdot p + \hat{t}\right)\ \forall p \in P'. \tag{8}$$

For each transformer decoder layer, we compute a cross entropy loss between the predicted similarity matrix (Eq. (4)) and the ground truth correspondence matrix $\hat{S}_{sim}$:

$$\mathcal{L}_{sim} = \sum_{l=1}^{L} \mathrm{CE}\left(\sigma\left(\frac{Q_l \cdot K^T}{d_{model}}\right), \hat{S}_{sim}\right), \tag{9}$$

with σ being the softmax operation. This formulation enforces a strong alignment between the attention-based similarity scores and the true 2D-3D associations, ensuring that each decoder layer progressively refines its understanding of pixel-point relationships.

Inspired by Li et al. (2025), we also introduce an auxiliary classification task to further stabilize the transformer decoder training. Specifically, during training, for each layer output, we also predict whether each point lies within the ground truth camera Field of View (FoV) and compute a binary cross entropy loss between the predicted scores and the ground truth one-hot vector $\hat{S}_{fov}$:

$$\mathcal{L}_{fov} = \sum_{l=i}^{L} \mathrm{BCE}\left(\theta\left(Q_l\right), \hat{S}_{fov}\right) \tag{10}$$

where θ is an MLP. As regard as the rotation and translation prediction heads, we employ an L1 loss $\mathcal{L}_{r\text{-}coarse}$, for supervising the sine and the cosine rotation values along each axis, while we minimize the L2 norm difference $\mathcal{L}_{t\text{-}coarse}$ between the predicted translation vector and the ground truth translation vector. Thus the Implicit Alignment optimization function is:

$$\mathcal{L}_{implicit} = \mathcal{L}_{r\text{-}coarse} + \mathcal{L}_{t\text{-}corse} + \mathcal{L}_{sim} + \mathcal{L}_{fov}. \tag{11}$$

3.3 BEV Alignment

Building upon the the idea proposed by Yuan et al. (2025), we choose to employ a Bird's Eye View (BEV) space to fuse the RGB and the LiDAR features. In order to accommodate rotation up to $\pm 360°$ as required by the standard benchmark, we further choose to formulate the Image-to-Point Cloud as a BEV alignment task, rather than leverage BEV features only for fusing the two input modalities. Following Yuan et al. (2025), we firstly aim to obtain two BEV representations of the same scene: one from the RGB image and the other from the sparse LiDAR point cloud. For this reason, inspired by object detection literature (Philion & Fidler, 2020; Harley et al., 2023), we first define a volume of 3D coordinates $G \in \mathbb{R}^{X\times Y\times Z\times 3}$ and a zero-initialized 3D space

${}^{3D}F_{rgb} \in \mathbb{R}^{X \times Y \times Z \times C}$, representing the 3D coordinates and the 3D features space obtained from the images surrounding the vehicle. We employ a East-Down-North (EDN) coordinate system, centering the volume with the camera frame origin. Then, in order to sample and lift the 2D features from F_{rgb} to ${}^{3D}F_{rgb}$, the 3D coordinates G are projected from the camera EDN frame $\{c\}$ onto the image plane with the intrinsics matrix K through Eq. (12) to get the corresponding 2D coordinates:

$$\begin{bmatrix} u & v & d \end{bmatrix}^T = K \cdot g, \ \forall g = \begin{bmatrix} {}^c x & {}^c y & {}^c z \end{bmatrix}^T \in G, \tag{12}$$

and we bilinearly sample the features given the 2D coordinates to fill the corresponding cells in ${}^{3D}F_{rgb}$.

Similarly, we define a zero-initialized 3D space ${}^{3D}F_{pts} \in \mathbb{R}^{X \times Y \times Z \times C}$. As previously described in Eq. (7), we leverage the predicted T_{coarse} from the Implicit Alignment module to coarsely align the point cloud coordinates P, which are then projected onto the three-dimensional space and the features F_{pts} are scattered into ${}^{3D}F_{pts}$ accordingly to the corresponding projection cells. Afterward, we flatten the Y and C axes both in ${}^{3D}F_{rgb}$ and ${}^{3D}F_{pts}$, obtaining BEV feature maps with dimensions $X \times Z \times (Y \cdot C)$. We then concatenate the two BEVs along the channels dimension, and feed it to a single 3×3 convolutional layer $\varphi(\cdot)$ followed by instance normalization and a non-linearity for fusing the two BEV modalities:

$${}^{3D}F_{fused} = \varphi\left({}^{3D}F_{rgb} \oplus {}^{3D}F_{pts}\right), \tag{13}$$

with $\oplus$ being the concatenation. Finally, the fused features are processed by a convolutional encoder which, thanks to its large receptive field, can reason both locally and globally about the geometry and the mis-alignment between the two BEVs, producing high-level features that benefit the rotation and translation heads. It is important to note that even without aligning the two BEVs, we empirically found that the encoder is sufficiently powerful to extract the necessary features to solve the calibration task. To this end, we adopt a ResNet-18 (He et al., 2016) architecture as the encoder, where the C5 features are further reduced with a single convolution layer before serving as input to the translation and rotation prediction heads. Once again, both heads are implemented as two-layer MLPs with ReLU non-linearity and a fine transformation matrix T_{fine} is decoded as in Eq. (6). Finally, the final calibration matrix T is obtained by combining the coarse and the fine calibration matrices:

$$T = T_{fine} \cdot T_{coarse} \tag{14}$$

As discussed in Section 1 and demonstrated by our experiments (see Section 4.4), the CNN-based encoder effectively aligns the concatenated feature maps ${}^{3D}F_{fused}$, yielding accurate calibration matrices. To this end, we supervise both the rotation and the translation heads as before. Thus, the BEV Alignment module optimization function is:

$$\mathcal{L}_{bev\text{-}align} = \mathcal{L}_r + \mathcal{L}_t. \tag{15}$$

Notably, both the Implicit and the BEV Alignment modules are optimized in an end-to-end fashion, avoiding any 2-step training strategy, as the one from Li et al. (2025). To this end, the overall optimization function results in the following:

$$\mathcal{L} = \mathcal{L}_{bev\text{-}align} + \mathcal{L}_{implicit}. \tag{16}$$

4 Experiments

4.1 Datasets and Metrics

We follow the standard benchmark in Image-to-Point Cloud registration task and we use the public KITTI (Geiger et al., 2012) and nuScenes (Caesar et al., 2020) datasets for training and evaluating our network. Regarding the KITTI dataset, we use the odometry benchmark which consists of 22 stereo sequences containing the images collected by two RGB cameras facing forward the vehicle and the point cloud acquired by a LiDAR mounted on top of the vehicle. Following previous works (Li & He Lee, 2021; Zhou et al., 2023; Ren et al., 2023; Lin et al., 2025; Li et al., 2025; Bie et al., 2025), we use the sequences from 00 to 08 for training and the sequences 09 and 10 for testing. In addition, we employ the images acquired by both the RGB stereo cameras both for training and testing. Instead, the nuScenes dataset consists of 1000 different scenes containing the images collected by 6 different cameras returning a 360° field of view of the scene surrounding the vehicle and the point clouds acquired by 5 radars and a LiDAR. Again, we follow previous works for train and test splits. Hence, we use the official 700 sequences in the train split and the 150 scenes in the evaluation split for training, while we use the remaining 150 of the test split for testing. Notably, we use only the images acquired by the front camera and the point clouds acquired by the LiDAR both for training and testing.

As evaluation metrics, again, we follow previous works (Li & He Lee, 2021; Zhou et al., 2023; Ren et al., 2023; Lin et al., 2025; Bie et al., 2025; Li et al., 2025) and we evaluate the registration performance with the average Relative Translation Error (RTE) and the average Relative Rotation Error (RRE) (Ma et al., 2016). Moreover, for a complete comparison, we also report the registration accuracy (Acc.) (Zhou et al., 2023) which corresponds to the proportions of registrations achieving RTE $<$ 2m and RRE $<$ 5°, even if it does not properly asses the fine-grained quality of predicted

Table 1 Registration results on the KITTI odometry and nuScenes datasets. Lower is better for both RTE and RRE, while higher is better for accuracy. Values are from original papers, except for BEVCalib which has been originally tested on a non-standard benchmark. '-' for missing results and training code.

	KITTI			nuScenes		
Method	RTE(m)↓	RRE(°)↓	Acc.↑	RTE(m)↓	RRE(°)↓	Acc.↑
Grid Cls. + PnP (Li & He Lee, 2021)	3.64 ± 3.46	19.19 ± 28.96	11.22	3.02 ± 2.40	12.66 ± 21.01	2.45
DeepI2P (3D) (Li & He Lee, 2021)	4.06 ± 3.54	24.73 ± 31.69	3.77	2.88 ± 2.12	20.65 ± 12.24	2.26
DeepI2P(2D) (Li & He Lee, 2021)	3.59 ± 3.21	11.66 ± 18.16	25.95	2.78 ± 1.99	4.80 ± 6.21	38.10
CorrI2P (Ren et al., 2023)	3.78 ± 65.16	5.89 ± 20.34	72.42	3.04 ± 60.76	3.73 ± 9.03	49.00
CurrI2P (CorrI2P) (Lin et al., 2025)	1.55 ± 7.99	3.61 ± 10.88	-	2.16 ± 4.20	2.98 ± 5.35	-
VP2P-Match (Zhou et al., 2023)	0.75 ± 1.13	3.29 ± 7.99	83.04	0.89 ± 1.44	2.15 ± 7.03	88.33
CurrI2P (VP2P-Match) (Lin et al., 2025)	0.53 ± 1.00	2.11 ± 9.48	-	1.04 ± 1.64	2.67 ± 8.61	-
RelaI2P (Hou et al., 2025)	0.72 ± 1.45	2.92 ± 6.67	85.60	-	-	-
ICLM (Li et al., 2025)	0.20 ± 0.21	1.24 ± 2.34	97.49	0.63 ± 0.44	2.13 ± 3.75	90.94
GraphI2P (Bie et al., 2025)	0.32 ± 0.81	1.65 ± 1.32	99.61	0.49 ± 1.22	1.73 ± 1.63	99.48
BEVCalib (Yuan et al., 2025)	0.88 ± 0.60	4.47 ± 8.79	74.28	-	-	-
Ours	**0.02 ± 0.01**	**0.32 ± 0.26**	**100.0**	**0.02 ± 0.01**	**0.35 ± 0.46**	**99.92**

registrations. Finally, we report the RTE and RRE standard deviations, computed by aggregating all the RTE and RRE values for each test sample.

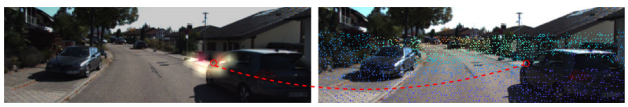


Fig. 3 Pixel-to-Point correspondences. For a given 3D point projected into the image plane (right), we highlight the highest scoring corresponding pixels (left)

4.2 Implementation Details

As done by Lin et al. (2025), Bie et al. (2025), Li et al. (2025), for the KITTI dataset, we set the input RGB dimensions to be 160×512, these are obtained by removing the top-50 rows, applying a 0.5 downsaple factor, and randomly cropping the resulting images to match the desired dimensions. Afterward, we downsample the the point cloud size to $N = 40960$. Similarly, for the nuScenes dataset, we follow the previous works by setting the images to be of size 160×320. These are obtained by removing the top-100 rows, applying a 0.2 downsample factor, and randomly cropping the resulting images to the desired dimensions. In regard to the 3D branch, we set the point cloud dimension $N = 40960$ by accumulating from the past frames as in any previous work, since a single LiDAR sweep in nuScenes contains roughly 36k points. However, this is not a required step. Hence, we refer to Section Section 4.4.6 for a further analysis.

In the Implicit Alignment module, we set the number of transformer decoder layers both in T_1 and T_2 to be 3, in order to follow the common practice to have a total number of 6 transformer layers. In T_1, we follow Li et al. (2025) to set the features dimension d_{model} to 128, while we employ single-head attention modules to correctly supervise the predicted similarity matrix (Eq. (4)). Conversely, in T_2, we set the features dimension to 512 and we employ multi-head attention modules with 8 attention heads.

In the BEV Alignment module, we set X, Y, and Z to 200, 8, 200, respectively, spanning a 3D metric space of ±25 in both the forward-backward and left-rigth directions and of ±5m in the up-bottom directions, with features dimension C to 128. Furthermore, we set the final convolution to have a kernel size of 7×7 with 512 output channels.

We set the batch size to 8 in all our experiments and train for 260k iterations using Adam (Kingma & Ba, 2014) as optimizer. We use 1×10^{-4} as initial learning rate, which is decayed by a 0.5 factor every 51k iterations.

4.3 Comparison with State-of-the-Art

In Table 1, we compare the performance of our framework with the latest state-of-the-art methods according to the standard benchmark, which consists in sampling a random rotation along the vertical axis in the range ±360° and a random translation along the forward-backward and left-right directions in the range ±10m. We also follow Zhou et al. (2023), Bie et al. (2025), Li et al. (2025) and do not remove the samples with large errors before computing the metrics, which would result in an unsuitable way to report the real performance. We refer to the appendix material for this specific evaluation. In order to ensure a fair comparison, we adopt the same experimental settings as all methods reported

Table 2 Ablation studies of each component. Implicit: Implicit Alignment module, $\mathcal{L}_{sim}$: effect of similarity loss between corresponding point-pixels pairs in the cross attention mechanism, BEV: BEV Alignment module.

			KITTI		nuScenes	
Implicit	$\mathcal{L}_{sim}$	BEV	RTE(m)↓	RRE(°)↓	RTE(m)↓	RRE(°)↓
			1.07	13.73	0.71	10.83
		✓	0.04	1.08	0.04	0.98
✓			1.90	7.57	1.88	86.97
✓	✓		0.2	1.42	0.19	2.42
✓	✓	✓	**0.02**	**0.32**	**0.02**	**0.35**

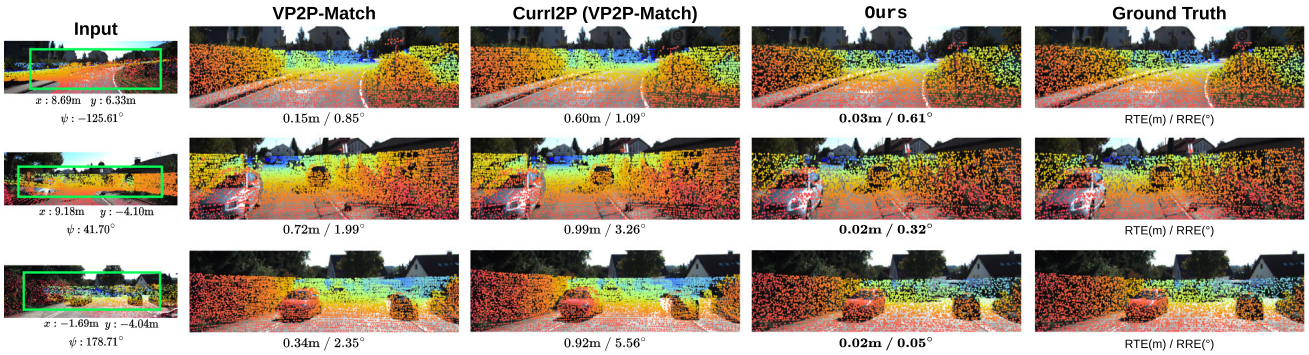


Fig. 4 Qualitative comparison of Image-to-Point Cloud registration result on the KITTI dataset. From left to right: mis-aligned point cloud and RGB inputs, VP2P-Match calibration result, CurrI2P (VP2P-Match) calibration result, our model calibration result, and the ground truth camera-LiDAR alignment

in Table 1 (see Section 4.2), including dataset sequences selection, camera choice, point cloud dimensionality, point cloud accumulation strategy, image resolution, and image pre-processing operations. Specifically, with the exception of BEVCalib, in Table 1 we report the registration performance directly from the original publications, as these methods already adhere to the standard experimental protocol. In contrast, we trained and evaluated BEVCalib using its publicly available implementation, configuring it consistently with our approach and the other baselines to ensure comparability. On the KITTI dataset, we outperform ICLM, which is to the best of our knowledge the current state-of-the-art model on the KITTI dataset, by 0.18m on the average RTE, by 0.92° on the average RRE and by 2.51% on the registration accuracy. We also highlight to be more robust compared to all previous works. Indeed, our proposal results in a reduction in the standard deviation of 0.20m and 2.08° on the RTE and RRE, respectively. Similarly, on the nuScenes dataset, we outperform the current state-of-the-art model GraphI2P by 0.47m, 1.38°, and 0.44% on the RTE, RRE, and registration accuracy, respectively. Once again, our method demonstrates superior robustness compared to all prior works, as it is able to reduce the standard deviation by 1.21m and by 1.17° on the RTE and RRE, respectively. Despite being a BEV-based approach, BEVCalib poorly performs under this benchmark. As matter of fact, compared to BEVCalib, our method achieves lower average RRE by 4.15° and lower average RTE by 0.86m. BEVCalib leverages the Feature Selector samples as a subset of BEV features which are subsequently processed via self-attention. As stated by the authors, this strategy inherently focuses on the overlapping region between camera and LiDAR, thus when a large camera-LiDAR misalignment occurs, as in this benchmark, the camera features acting as queries may not sufficiently correspond to the scattered LiDAR representations within the BEV space. Since the sampled BEV features are generated by an FPN-based encoder with a limited receptive field, only a minority of sampled tokens participate effectively in the attention mechanism, leading to weak cross-modal associations and degraded calibration performance compared to methods that preserve broader or more globally consistent feature correspondences.

4.4 Ablation Studies

In this section, we conduct ablation studies to highlight how different parameters and architectural choices reflect on the registration task. We report the effectiveness of each contribution of our work in Table 2. Inspired by Li and He Lee (2021), we test a baseline architecture just composed by the modality-specific backbones, recovering one single global feature from the RGB and one from the point cloud, directly regressing the calibration from their concatenation. This architecture suffers from a large error in both rotation

Table 3 Analysis on our method generalization ability when trained on left camera images and tested on right ones.

		KITTI		
Train	Test	RTE(m)↓	RRE(°)↓	Acc.↑
L+R	R	0.02 ± 0.01	0.31 ± 0.26	100.0
L	R	0.03 ± 0.01	0.68 ± 0.46	99.98

and translation, caused by the lack of spatial information, serving only as a reference with respect to further methods. Then, we ablate the effect of our BEV Alignment module (second row of Table 2), where we project both the RGB and the cloud features into a BEV space. We highlight that our first contribution already outperforms the other state-of-the-art models both on KITTI and nuScenes. Next, we test our Implicit Alignment module alone to gain spatial information by cross-attending the LiDAR cloud with the RGB image. As shown in the third row of Table 2, this module struggles to find the correct calibration matrix on the KITTI dataset, while it fails to converge on the nuScenes dataset. We argue that this is due to the instability of the transformer training, demonstrating the requirement for a stronger supervision. Afterward, by supervising the attention matrix computed through cross-attention between 3D and 2D features to correctly estimate the point-to-pixel correspondences, we are able to achieve comparable results to the state-of-the-art models both on KITTI and nuScenes (fourth row of Table 2). We argue that this result shows the ability of this loss in stabilizing the transformer training, leading to better calibration performance by simplifying the prediction heads task. We further illustrate the effect of $\mathcal{L}_{sim}$ in Fig. 3, where a selected 3D point (circled in red, bottom figure) is projected onto the image plane for visualization. The best correspondence scoring pixels in features space are highlighted on the image (top figure). As shown, for a 3D point belonging to the car on the right side, the highest-scoring pixels consistently lie within the car region, demonstrating the correctness of the similarity matrix output by cross-attending the 3D features with the ones from the RGB branch. Lastly, we leverage the coarse estimate from the Implicit Alignment module to enhance the BEV Alignment module registration performance, where we project both the RGB and the coarsely aligned cloud features into the BEV space. This contribution leads to an RRE reduction of 0.76° and 0.63° on KITTI and nuScenes, respectively (last row of Table 2), compared to the BEV Alignment alone, supporting our choice to exploit the re-alignment of features at spatial level through the BEV representation.

4.4.1 Generalization Analysis

Memorization of a particular sensor suite constitutes an actual issue when training a model that relies on strong geometrical representation like the Bird's Eye View. The slightest rig modification could potentially harm the registration performance, resulting in an unreliable calibration pipeline. For this reason, we decided to introduce a controlled variable into the performance evaluation, by modifying the camera's spatial layout between the training and testing phases. Therefore testing if our model memorized a fixed sensor configuration. We stressed this generalization capability by training the model solely on the left camera images and evaluating on the right camera ones on the KITTI dataset. In Table 3, we show how our network performs given a different sensor configuration between training and testing. Indeed, the registration performance does not get undermined when the model is trained solely on the left camera images and evaluated on the right camera ones, compared to the model trained on both the camera images and evaluated on the right ones. We argue that a slight degradation of the registration performance might be expected by the Implicit Alignment, because of the smaller training set. However, even with this slight degradation when training the our model solely on the left camera images, we still outperform other state-of-the-art models trained on both camera images.

4.4.2 6DoF Registration Analysis

Although this is not the standard benchmark for point-based methods, in Table 4 we show our network registration performance on the KITTI dataset, in a Six Degrees of Freedom (6DoF) scenario, where variations for the three spatial coordinates and rotation angles are involved. Following projection-based methods, such as Lv et al. (2021) and Iyer et al. (2018), for each axis we sample both a random rotation and a random translation in range ±20° and ±1.5m, respectively. Our model outperforms the latest state-of-the-art point-based approaches with publicly available training code CorrI2P and BEVCalib. Specifically, we outperform CorrI2P by 0.91m, 2.34°, and 14.24% on the RTE, RRE, and registration accuracy, respectively, while BEVCalib by 0.02m and 0.10° on the RTE and the RRE, respectively.

For a fair and complete analysis, we also compare out model against Calibnet (Iyer et al., 2018) and LCCNet (Lv et al., 2021), two state-of-the-art projection-based approaches. Specifically, we outperform Calibnet by 5.83m, 10.39°, and 96.97% on the RTE, RRE, and registration accuracy, respectively, while we surpass LCCNet by 0.35m, 3.74°, and 24.25%. Once again, our approach shows to be more robust compared to all previous works that have been tested.

4.4.3 RGB Lifting Strategy

In Table 5 we compare different RGB features lifting strategies. Specifically, we compare two different strategies: the one employed by Yuan et al. (2025) and proposed by Phil-

Table 4 6DoF registration analysis. For each axis sample a rotation between ±20° and a translation between ±1.5m.

Method	KITTI		
	RTE(m)↓	RRE(°)↓	Acc.↑
CorrI2P (Ren et al., 2023)	0.96 ± 3.10	2.87 ± 4.58	85.76
BEVCalib (Yuan et al., 2025)	0.07 ± 0.04	0.63 ± 0.38	**100.0**
Calibnet (Iyer et al., 2018)	5.88 ± 2.80	10.92 ± 6.09	3.03
LCCNet (Lv et al., 2021)	0.40 ± 0.29	4.27 ± 3.70	75.75
Ours	**0.05 ± 0.03**	**0.53 ± 0.35**	**100.0**

Table 5 Comparison against different RGB lifting strategies.

Method	KITTI		
	RTE(m)↓	RRE(°)↓	Acc.↑
Lift-Splat (Philion & Fidler, 2020)	**0.02 ± 0.01**	0.33 ± 0.27	99.98
SimpleBEV (Harley et al., 2023)	**0.02 ± 0.01**	**0.32 ± 0.26**	**100.0**

Table 6 Comparison against different RGB backbones. VP2P-Match results from the original paper.

Method	Backbone	KITTI		
		RTE(m)↓	RRE(°)↓	Acc.↑
VP2P-Match (Zhou et al., 2023)	Res-34	0.75 ± 1.13	3.29 ± 7.99	83.04
BEVCalib (Yuan et al., 2025)	Swin-T (Liu et al., 2021)	0.88 ± 0.60	4.47 ± 8.79	74.28
Ours	Res-34	0.02 ± 0.02	0.33 ± 0.64	99.96
Ours	Res-50	**0.02 ± 0.01**	0.32 ± 0.26	**100.0**
Ours	Swin-T (Liu et al., 2021)	**0.02 ± 0.01**	**0.28 ± 0.27**	99.98

ion and Fidler (2020), and the one proposed by Harley et al. (2023). Thus, we demonstrate that our method results to be agnostic to the chosen lifting strategy, since both the tested strategies lead to similar registration performance.

4.4.4 2D Backbone Analysis

Table 6 shows the registration performance of our model against different RGB backbones. Specifically, our network outperforms both VP2P-Match and BEVCalib even when employing the same RGB backbone. We also highlight that BEVCalib uses a different input RGB size compared to the other state-of-the-art models, which is set to be 375 × 1242. For this reason, we employ the same input resolution as BEVCalib when comparing to them, while we opt for a standard input resolution when comparing against the other state-of-the-art models.

4.4.5 BEV Resolution Analysis

In Table 7 we compare different 3D grid resolutions. We found an optimal setup by setting the cell dimensions to 0.25m along the forward-backward and left-right directions. Specifically, we limit the BEV dimensions to 200×200, since higher resolution would require greater memory requirements.

Table 7 BEV hyperparameters analysis.

BEV	Range(m)	Cell(m)	KITTI		
			RTE(m)↓	RRE(°)↓	Acc.↑
200 × 200	±100	1.0	0.08	1.31	98.97
200 × 200	±75	0.75	0.06	0.94	99.53
200 × 200	±50	0.50	0.03	0.64	99.94
200 × 200	±25	0.25	**0.02**	**0.32**	**100.0**

4.4.6 Point Cloud Accumulation Analysis

As discussed in Section 4.2, accumulating multiple point cloud sweeps is not strictly necessary for our model to outperform prior approaches. Accordingly, unlike previous works (Bie et al., 2025; Li et al., 2025; Lin et al., 2025; Zhou et al., 2023) which set the point cloud dimension to $N = 40960$ even on nuScenes dataset, in this section we downsample the point cloud to $N = 36000$, as a single LiDAR sweep in nuScenes contains roughly 36k points. This should be a required step since, in this setting, we do not apply any accumulation strategy over past and future frames. We argue that

Table 8 Registration results against latest state-of-the-art approaches without using point cloud accumulation techniques on the nuScenes dataset.

Method	nuScenes		
	RTE(m)↓	RRE(°)↓	Acc.↑
VP2P-Match (Lin et al., 2025)	0.89 ± 1.44	2.15 ± 7.03	88.33
CurrI2P(VP2P-Match) (Lin et al., 2025)	1.04 ± 1.64	2.67 ± 8.61	-
ICLM (Li et al., 2025)	0.63 ± 0.44	2.13 ± 3.75	90.94
GraphI2P (Bie et al., 2025)	0.49 ± 1.22	1.73 ± 1.63	99.48
Ours w/o acc.	**0.02 ± 0.01**	**0.29 ± 0.22**	**100.0**
Ours	**0.02 ± 0.01**	0.35 ± 0.46	99.92

accumulating future frame hinders the applicability of the method in real-world scenarios, while accumulation over past frames typically requires accurate ego-poses information to be available. Despite not leveraging point cloud accumulation, Table 8 shows that our model achieves superior registration performance compared to all the previous state-of-the-art methods.

4.4.7 Inference Time Analysis

In Table 9, we report both the number of parameters and the inference time of our method compared to the latest works with a publicly available code, measured on an NVIDIA L40S with a fixed batch size of 1. To explicitly analyze the trade-off between inference speed and registration performance, we evaluate our approach using different 3D backbone configurations, including both a faster and a slower variant of PointNet++, as well as the more recent PointTransformerV3 (Wu et al., 2024). We refer to the appendix material for a detailed description of the differences between the standard and the fast PointNet++. When employing the slower PointNet++ backbone, our model outperforms by 96% and 84% CurrI2P (VP2P-Match) in terms of RTE and RRE, respectively, at the cost of an additional 59.5ms of inference time. Conversely, when using the faster PointNet++ configuration, our approach achieves improvement of 94% and 80% over CurrI2P (VP2P-Match) on the RTE and RRE, respectively, while reducing the inference time by 141.8ms. Compared to the fastest baseline BEVCalib, our method is 201.3ms slower, but it achieves substantial error reductions of 97% and 92% in the RTE and RRE, respectively. Notably, when adopting the faster PointNet++ backbone, this gap is reduced to only 53.4ms, without incurring in any significant degradation in registration performance. Finally, replacing PointNet++ with the more recent PointTransformerV3 backbone reduces the inference time by 171.6ms compared to our slowest configuration, while maintaining comparable registration accuracy once again. Overall, our approach achieves an inference time below 0.5s in its most accurate configuration and under 100ms in its fastest one, while maintaining competitive registration performance. These results demonstrate that our method is suitable for an online deployment scenarios.

4.4.8 Qualitative Analysis

A qualitative analysis on the KITTI dataset is shown in Fig. 4. For visualization, we first apply the predicted alignment transformation matrix to the point cloud, which is then projected onto the image plane through the known camera intrinsics parameters. We compare our method (fourth column from left) against the state-of-the-art models CurrI2P and VP2P-Match, and the ground truth (last column from left), reporting both the RTE and the RRE for each prediction. Instead, in the first column, we show the mis-registered image and point cloud inputs. Notably, we highlight the ability of our method to achieve better alignment performance compared to CurrI2P in complex scenarios, in which the network might struggle to extract distinctive visual features. Indeed, by leveraging the Implicit Alignment module and the explicitly aligned BEV features, our model demonstrates to have a better understanding of the geometrical relationships in the surrounding environment. Finally, we refer to the appendix for a further analysis on the nuScenes dataset, on the failure cases, and on the challenging weather conditions.

5 Conclusion

In this paper, we introduced a framework that casts the LiDAR-Camera calibration task as a BEV alignment task. Taking inspiration from previous BEV-based work, our key insight is to extract 3D BEV representations from both the modalities and directly estimate the calibration matrix by aligning these representation through a convolutional encoder. Then, we further proposed an Implicit Alignment step to directly cross attend 2D and 3D features to estimate a coarse calibration matrix for explicitly align the BEV features and refine the calibration matrix. Extensive experiments demonstrate that our approach achieves state-of-the-art performance, surpassing previous methods by a significant margin in terms of accuracy and robustness. Future works

Table 9 Number of parameters (M) and inference time (ms) on a NVIDIA L40S for each method with publicly available code. Our approach is evaluated using different 3D backbones to analyze the impact of model size and inference time on registration performance. PN++ corresponds to the PointNet++ backbone (Qi et al., 2017b), while PTv3 stands for the PointTransformerV3 backbone (Wu et al., 2024).

Method	# params.(M)	Time(ms)↓	KITTI	
			RTE(m)↓	RRE(°)↓
VP2P-Match	36.6	226.2	0.75	3.29
CurrI2P (VP2P-Match)	36.6	236.5	0.53	2.11
BEVCalib	47.4	**41.3**	0.88	4.47
Ours - PN++	60.4	296.0	**0.02**	**0.32**
Ours - PN++ (small)	60.4	94.7	0.03	0.41
Ours - PTv3	94.8	124.4	0.04	0.42

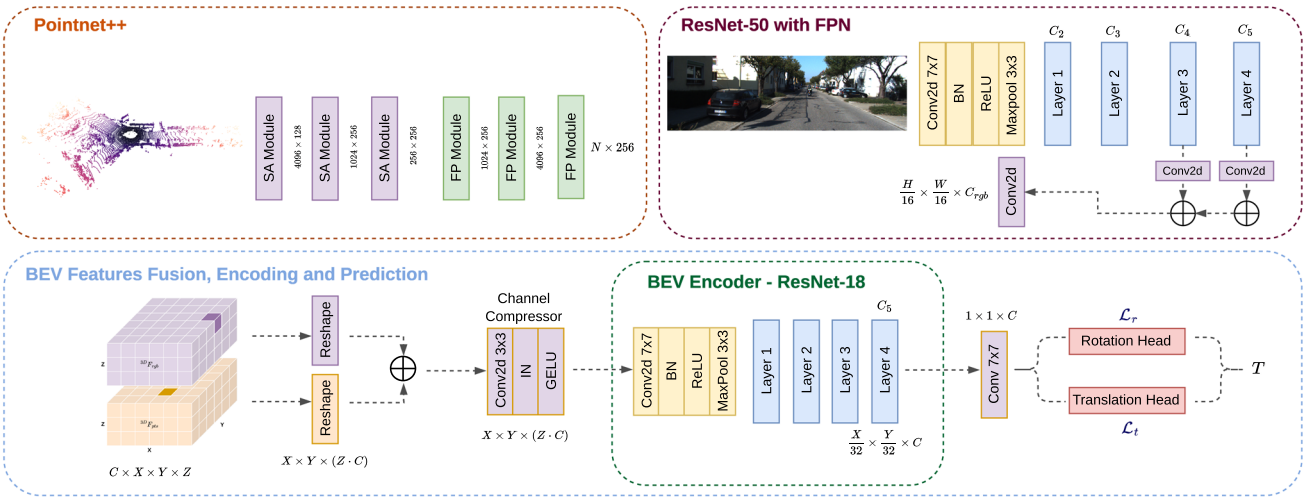


Fig. 5 Images and point clouds as processed to extract two different *Bird's Eye Views* (BEVs) representations of the same 3D space surrounding the vehicle. The two BEVs are then fused by concatenation and processed by a final CNN-based encoder to predict the calibration matrix that aligns the two input modalities

may target architectural changes, for example, by avoiding the use of any CNN-based encoder used for the two BEVs feature fusion, in favor of a sparse module. Furthermore, expanding the model to process multi-camera information could be an effective strategy to enrich and densify the lifted BEV ${}^{3D}F_{rgb}$ feature map, with additional visual information, leading to better registration performance.

Appendix A Network Details

In this section, we dive into the details of our architectural choices, first analyzing the modality-specific backbones, followed by an in-depth look at how the two-step alignment algorithm works with a further description.

A.1 Backbones

As point cloud backbone we employ a customized version of the PointNet++ backbone (Qi et al., 2017a, b, 2019) in an encoder-decoder fashion, employing 3 downsample and upsample blocks. In the standard configuration, the input point cloud containing N points is progressively downsampled to 4096 in the first stage, to 1024 in the second stage, and to 256 points in the third downsampling block. Subsequently, the resulting 256-point representation is upsampled to 1024 points in the first upsampling stage, to 4096 points in the second stage, and finally back to N points in the last stage. In the faster configuration, the input point cloud is downsampled to 1024 in the first stage, to 512 points in the second stage, and to 256 in the third downsampling block. The upsampling path mirrors this design once again, expanding the 256-point representation to 512 points in the first upsampling stage, to 1024 points in the second stage, and finally to N points in the last stage. Lastly, as shown in Fig. 5 (left), the 3D features

computed in the first upsampling stage serves as $F^{'}_{pts}$. As regard as the RGB backbone, we depict in Fig. 5 (right) the FPN architecture we employed in our framework.

A.2 Two-step Alignment Algorithm

Our two-step alignment algorithm is designed to exploit an intermediate coarse alignment, akin the Implicit Alignment, for simplifying the BEV Alignment task. Algorithm 1 highlights the end-to-end training pipeline that optimizes both the Implicit Alignment and the BEV Alignment modules.

Algorithm 1 Training

for $i \leftarrow 0$ to N **do** ▷ N number of iterations
  $I, P, T_{GT} \leftarrow$ Dataset[i]
  $F_{rgb} \leftarrow$ Conv1x1(Backbone2D(I))
  $F^{'}_{pts}, F_{pts} \leftarrow$ Backbone3D(P)
  $F^{'}_{pts}, F_{pts} \leftarrow$ Linear($F^{'}_{pts}$), Linear(F_{pts})
  $T_{coarse} \leftarrow$ ImplicitAlignment($F_{rgb}, F^{'}_{pts}$)
  $T_{fine} \leftarrow$ BEVAlignment($F_{rgb}, F_{pts}, T_{coarse}$)
  $T \leftarrow T_{fine} \cdot T_{coarse}$
  $\mathcal{L} \leftarrow \mathcal{L}_r(T, T_{GT}) + \mathcal{L}_t(T, T_{GT}) + \mathcal{L}_{sim}$

  $\mathcal{L}$.backward()
end for

Appendix B Additional Experiments

B.1 Metrics Formulation

In the following lines, we show in detail what the Relative Rotation Error (RRE) and the Relative Translation Error (RTE) represent, and how they are computed (Ma et al., 2016). The RRE is computed by converting the relative rotation between the ground truth rotation and the predicted one into Euler angles, which are then summed together:

$$\begin{aligned} \text{RRE} &= \sum_{i=1}^{3} |ypr(i)|\,, \\ ypr &= f\left(\hat{R}^{-1} \cdot R\right), \end{aligned} \tag{B1}$$

where $\hat{R}$ is the ground truth rotation matrix, R is the estimated rotation matrix from the predicted T, and $f(\cdot)$ transforms a rotation matrix into the three Euler angles. The RTE represents the magnitude of the difference between the ground-truth translation t_{GT} and the predicted translation t_{pred}:

$$\text{RTE} = \|t_{GT} - t_{pred}\|_2. \tag{B2}$$

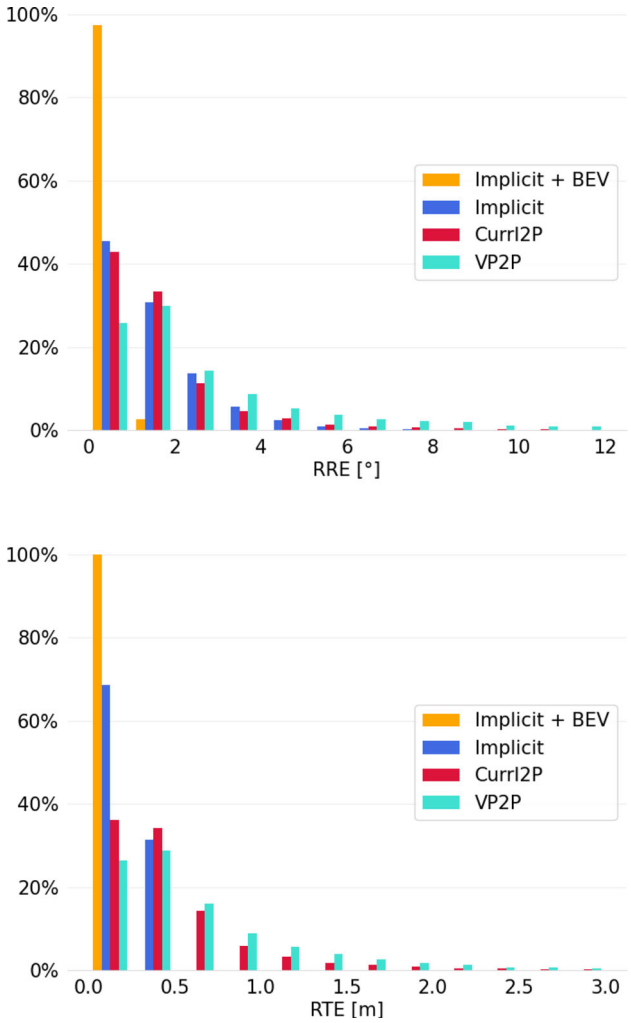


Fig. 6 Histograms of image-point cloud registration RRE and RTE on the KITTI dataset. x-axis is RRE (°) and RTE (m), y-axis is the percentage

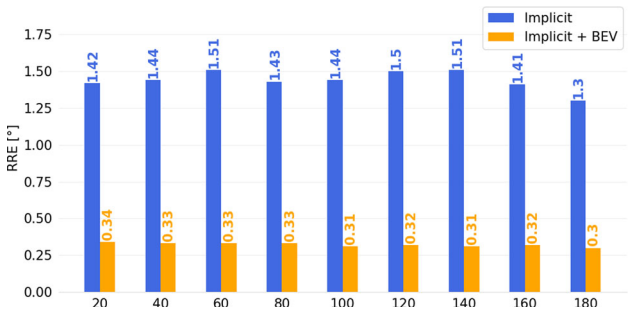


Fig. 7 Rotation error for different initial mis-calibration on the KITTI dataset. x-axis is the initial mis-calibration interval in degrees, y-axis is the RRE (°) computed from all the samples that falls into the specific interval

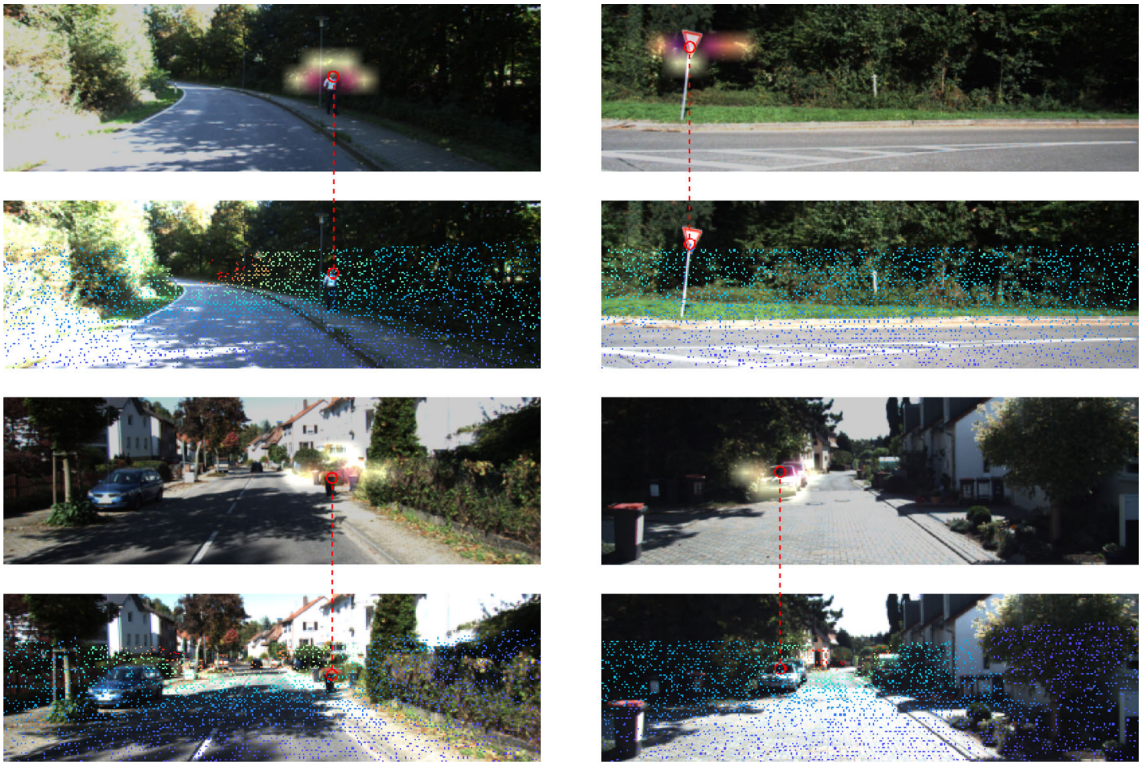


Fig. 8 3D-2D correspondences. For each couple of images, for a given 3D point projected into the image plane (bottom), we highlight the best-scoring correspondent pixels (above)

Table 10 Registration results on the KITTI odometry and nuScenes datasets. Lower is better for both RTE and RRE, while higher is better for accuracy. Results are filtered considering only samples with RTE $< 5m$ and RRE $< 10°$. '-' due to missing publicly available code.

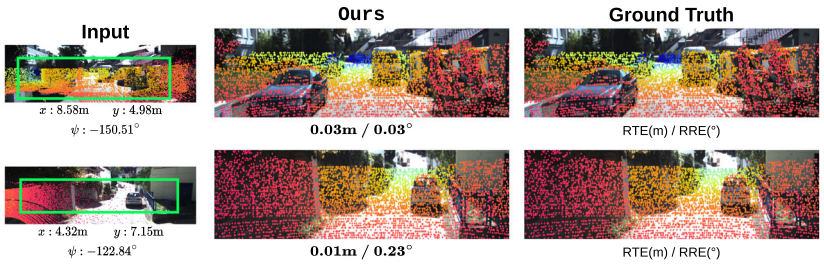
	KITTI (< 5m & $< 10°$)			nuScenes (< 5m & $< 10°$)		
Method	RTE(m)↓	RRE(°)↓	Acc.↑	RTE(m)↓	RRE(°)↓	Acc.↑
Grid Cls. + PnP (Li & He Lee, 2021)	1.07 ± 0.61	6.48 ± 1.66	49.67	2.35 ± 1.12	7.20 ± 1.65	58.90
DeepI2P (3D) (Li & He Lee, 2021)	1.27 ± 0.80	6.26 ± 2.29	51.46	2.00 ± 1.08	7.18 ± 1.92	17.02
DeepI2P(2D) (Li & He Lee, 2021)	1.46 ± 0.96	4.27 ± 2.74	59.98	2.19 ± 1.16	3.54 ± 2.51	83.50
CorrI2P (Ren et al., 2023)	0.74 ± 0.65	2.07 ± 1.64	89.06	1.83 ± 1.06	2.65 ± 1.93	89.30
VP2P-Match (Zhou et al., 2023)	0.59 ± 0.57	2.39 ± 2.07	95.40	0.73 ± 0.65	1.39 ± 1.63	97.15
CurrI2P (VP2P-Match) (Lin et al., 2025)	$\underline{0.44} \pm 0.42$	$\underline{1.39} \pm 1.44$	<u>95.98</u>	-	-	-
BEVCalib (Yuan et al., 2025)	0.85 ± 0.58	3.48 ± 2.21	77.91	-	-	-
Ours	$\mathbf{0.02 \pm 0.01}$	$\mathbf{0.32 \pm 0.26}$	**100.0**	$\mathbf{0.02 \pm 0.01}$	$\mathbf{0.33 \pm 0.27}$	**99.98**

B.2 Large Error Removal Testing

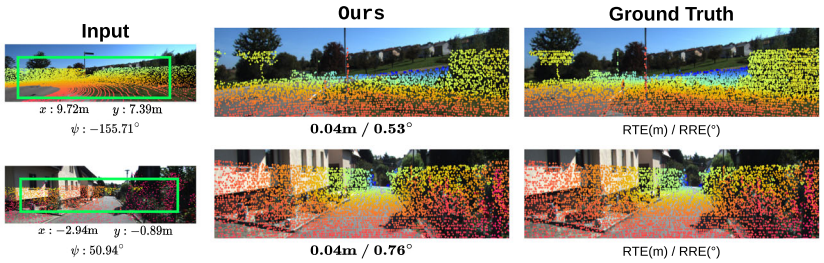
In order to avoid large errors due to failed registration, previous works, such as Ren et al. (2023), Li and He Lee (2021), compute the average RTE and RRE only for those samples with RTE lower than 5m and RRE lower than 10°. A comparison against previous state-of-the-art methods in this specific setting is shown in Table 10. Once again, our model outperforms all the state-of-the-art methods by a significant margin, while yielding a robuster predictions. We report once again the accuracy metric for completeness, although we argue that it is not a significant metric in the scenario described above.

B.3 Error Analysis

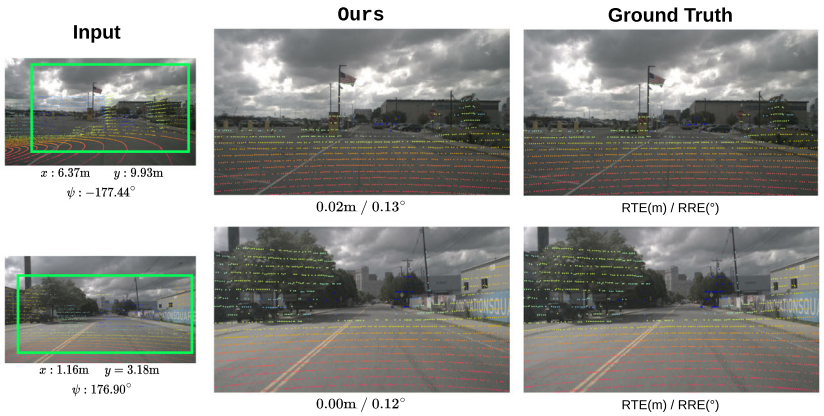
In this section, we evaluate the performance of against CurrI2P (Lin et al., 2025) and VP2P-Match (Zhou et al., 2023) across different mis-registration ranges in Fig. 6, reporting the error distribution in terms of RRE and RTE. The results show that CurrI2P and inherently VP2P-Match exhibits a longer tail in both distributions, with errors reach-



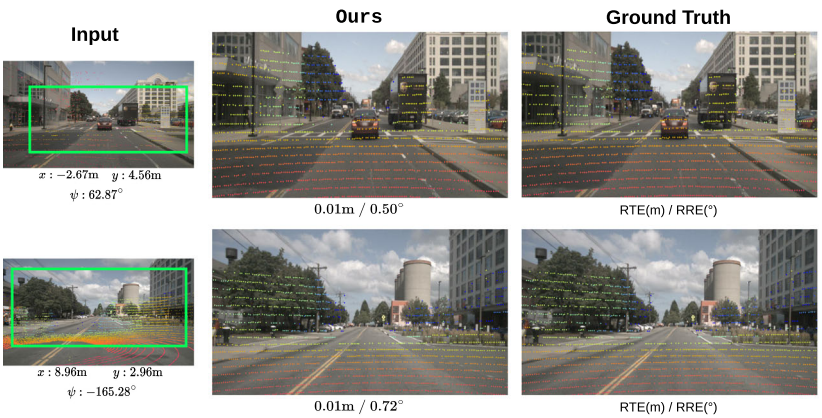
(a) Qualitative comparison of Image-to-Point Cloud registration results on the KITTI dataset.



(b) Examples of failure cases of Image-to-Point Cloud registration on the KITTI dataset

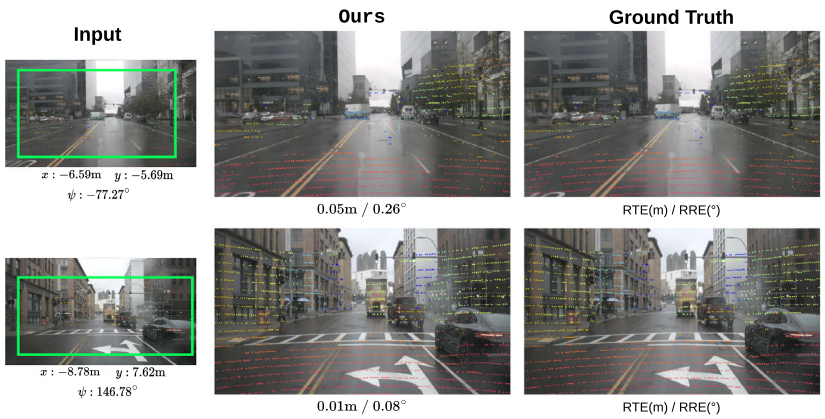


(c) Qualitative comparison of Image-to-Point Cloud registration results on the nuScenes dataset.

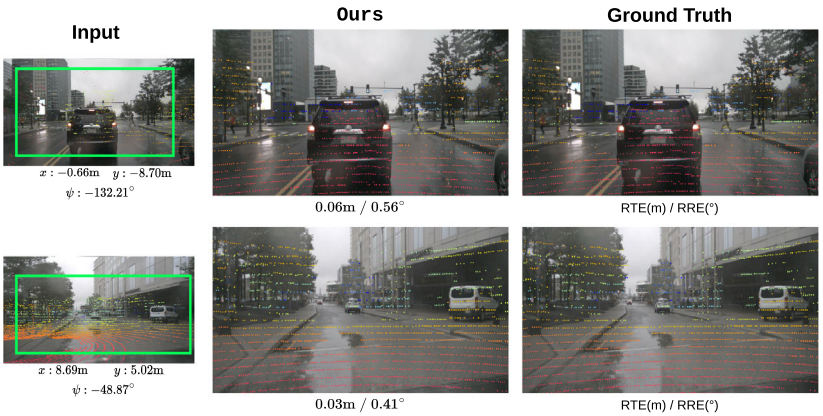


(d) Examples of failure cases of Image-to-Point Cloud registration on the nuScenes dataset

Fig. 9 KITTI and nuScenes qualitative analysis. From left to right: mis-aligned point cloud and image inputs, our model calibration result, and the ground truth camera-LiDAR alignment. For each prediction, we report both RRE and RTE



(a) Qualitative comparison of Image-to-Point Cloud registration results on the nuScenes dataset in rainy weather conditions.



(b) Examples of failure cases of Image-to-Point Cloud registration on the nuScenes dataset in rainy weather conditions.

Fig. 10 nuScenes qualitative analysis in adverse weather conditions. From left to right: mis-aligned point cloud and image inputs, our model calibration result, and the ground truth camera-LiDAR alignment. For each prediction, we report both RRE and RTE

ing up to 12° for RRE and 3m for RTE, indicating low robustness to large mis-registrations. In contrast, our Implicit Alignment module significantly narrows the error distribution, effectively mitigating larger calibration errors, which is an essential property for online applications. The proposed BEV Alignment is a step further to enhance robustness, yielding a more compact error distribution. To further demonstrate the effectiveness of our BEV Alignment module, we analyze the mean error for different initial mis-registration intervals. As shown in Fig. 7 we gain a great advantage from the BEV Alignment module in terms of RRE, as the new decoder is facilitated by the explicit alignment between BEVs features.

B.4 Similarity Loss Effect

In Fig. 8, we present an additional display of images that further emphasize the significance of the similarity loss in showing the best-scoring 3D-2D correspondences. This visualization aims to provide a clearer understanding of how the proposed similarity loss plays a pivotal role in aligning these two distinct modalities. For example, we show how similar the features of different objects are, respectively, represented by pixel features and point features.

B.5 Qualitative Self-Assessment Analysis

We conduct a qualitative self-assessment analysis both on the KITTI Odometry (Geiger et al., 2012) and the nuScenes (Caesar et al., 2020) datasets to better understand the model behavior. Indeed, in Fig. 9 we show some qualitative examples by firstly re-aligning the mis-registered point cloud through the predicted calibration matrix and then projecting it into the image plane via the known intrinsics parameters. Fig. 9(a) shows two samples from the KITTI dataset in which our model achieves excellent results, whereas, Fig. 9(b) shows two different circumstances in which the model outputs an inaccurate registration matrix. Similarly, Fig. 9(c) shows some samples in which our network exhibits almost perfect registration performance on the nuScenes dataset. However, for a fair and complete analysis on the nuScenes dataset, Fig. 9(d) shows two samples in which the model highlights non-negligible rotation errors in the predictions.

B.6 Adverse Condition Qualitative Self-Assessment Analysis

We also conduct a qualitative self-assessment on nuScenes under adverse weather conditions to better analyze potential limitations of our proposal. In Fig. 10(a) we show two samples in which the model achieves excellent results. However, Fig. 10(b) shows two samples in which our model highlights non-negligible rotation errors. Generally, we observe a higher mean error under adverse weather conditions, which we attribute to potential image distortions caused by raindrops directly impacting the camera lens. In such scenarios, our a multi-camera approach may provide richer RGB information, thereby enhancing registration performance.

B.7 Cross-Domain Generalization

Domain Adaptation (DA) is required when introducing a shift at both geometrical and semantic levels, such as a change of data between the training a testing sets. As demonstrated in Section Section 4.4.1, training on a given camera and testing on a different one remains feasible because both sensors belong to the same acquisition domain and share most latent properties. Our model successfully learns geometry-aware correspondences between LiDAR and RGB image, avoiding the memorization of a single camera's spatial layout. On the other hand, cross-domain generalization is a broader task which introduces additional variables, such as different camera projection models, LiDAR sampling patterns with peculiar noise characteristics, mounting configurations and environmental distributions. Therefore, evaluating transfer between multiple cameras within the same sensor suite provides a controlled setting, in which environmental and sensing statistics remain constant, while the extrinsic transformation changes. With this protocol, we directly test whether our model learns modalities-consistent geometry or memorizes a fixed sensor configuration. Cross-domain calibration is an important future direction, requiring dedicated mechanisms for handling simultaneous camera and LiDAR domain shifts.

Author Contributions All authors contributed to the study conception and design. Implementation and experimental analysis were performed by Cipelli Lorenzo and D'Addeo Filippo. The first draft of the manuscript was written by D'Addeo Filippo and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open access funding provided by Alma Mater Studiorum - Università di Bologna within the CRUI-CARE Agreement. No funding was received to assist with the preparation of this manuscript.

Data Availability The KITTI Odometry and the nuScenes dataset employed in this article can be found at https://www.cvlibs.net/datasets/kitti/eval_odometry.php and https://www.nuscenes.org/nuscenes\#download, respectively.

Declarations

Conflicts of Interest The authors have no relevant financial or non-financial interests to disclose.

Ethical Approval This study did not involve human participants or animals.

Consent to Participate Not applicable.

Consent for Publication Not applicable.

References

Bertogalli, A., Boracchi, G., & Magri, L. (2025). One target to align them all: Lidar, rgb and event cameras extrinsic calibration for autonomous driving arXiv:2511.12291.

Bhat, S. F., Birkl, R., Wofk, D., Wonka, P., & Müller, M. (2023). Zoedepth: Zero-shot transfer by combining relative and metric depth arXiv:2302.12288.

Bie, L., Pan, S., Li, S., Zhao, Y., & Gao, Y. (2025). Graphi2p: Image-to-point cloud registration with exploring pattern of correspondence via graph learning. *Proceedings of the Computer Vision and Pattern Recognition Conference* (pp. 22161–22171).

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., & Beijbom, O. (2020). nuscenes: A multimodal dataset for autonomous driving. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 11621–11631).

Chen, H., Wang, P., Wang, F., Tian, W., Xiong, L., & Li, H. (2022). Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation. arXiv:2203.13254.

Cheng, L., Sengupta, A., & Cao, S.*2023) 3d radar and camera co-calibration: A flexible and accurate method for target-based extrinsic calibration*. arXiv:2307.15264.

Darcet, T., Oquab, M., Mairal, J., & Bojanowski, P. (2023). Vision transformers need registers arXiv:2309.16588.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. *2012 IEEE conference on computer vision and pattern recognition* (pp. 3354–3361). IEEE.

Harley, A. W., Fang, Z., Li, J., Ambrus, R., & Fragkiadaki, K. (2023). Simple-BEV: What really matters for multi-sensor bev perception? *IEEE International Conference on Robotics and Automation (ICRA)*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 770–778).

Hou, M., Wang, G., Wang, Z., & Ma, B. (2025). Relai2p: Relational learning for image-to-point cloud registration. In: *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, (pp. 1–5).

Huang, Z., Zhang, Y., Chen, Q., & Fan, R. (2024). Online, target-free lidar-camera extrinsic calibration via cross-modal mask matching. *IEEE Transactions on Intelligent Vehicles, 10*, 3531–3542.

Iyer, G., Ram, R. K., Murthy, J. K., & Krishna, K. M. (2018). Calibnet: Geometrically supervised extrinsic calibration using 3d spatial transformer networks. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, https://doi.org/10.1109/iros.2018.8593693,

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization arXiv:1412.6980.

Koide, K., Oishi, S., Yokozuka, M., & Banno. A (2023) General, single-shot, target-less, and automatic lidar-camera extrinsic calibration toolbox. *2023 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 11301–11307). IEEE.

Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2019). Pointpillars: Fast encoders for object detection from point clouds. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12697–12705).

Lepetit, V., Moreno-Noguer, F., & Fua, P. (2009). Epnp: An accurate o(n) solution to the pnp problem. https://doi.org/10.1007/s11263-008-0152-6, https://infoscience.epfl.ch/handle/20.500.14299/60530.

Levinson, J., & Thrun, S. (2013). Automatic online calibration of cameras and lasers. In: *Robotics: Science and Systems*, https://api.semanticscholar.org/CorpusID:10004324.

Li, J., & Hee Lee, G. (2021). Deepi2p: Image-to-point cloud registration via deep classification. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, (pp. 15955-15964). https://doi.org/10.1109/cvpr46437.2021.01570.

Li, J., Chen, B. M., & Lee, G. H. (2018). So-net: Self-organizing network for point cloud analysis arXiv:1803.04249.

Li, X., Yang, W., Deng, J., Cheng, Z., Zhou, X., & Zhang, T. (2025). Implicit correspondence learning for image-to-point cloud registration. *Proceedings of the Computer Vision and Pattern Recognition Conference* (pp. 16922–16931).

Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., & Dai, J. (2024). Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 47*, 2020–2036.

Lin, L., Lin, C., Nie, L., Huang, S., & Zhao, Y. (2025). Curri2p: inter-and intra-modality similarity curriculum learning for image-to-point cloud registration. The Visual Computer (pp. 1–14).

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117–2125).

Liu, Y., Wang, T., Zhang, X., & Sun, J. (2022). Petr: Position embedding transformation for multi-view 3d object detection. *European conference on computer vision* (pp. 531–548). Springer.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE/CVF international conference on computer vision*, (pp. 10012–10022).

Lv, X., Wang, B., Ye, D., & Wang, S. (2021). Lccnet: Lidar and camera self-calibration using cost volume network. arXiv:2012.13901.

Ma, Y., Guo, Y., Zhao, J., Lu, M., Zhang, J., & Wan, J. (2016). Fast and accurate registration of structured point clouds with small overlaps. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1–9).

Pandey, G., McBride, J., Savarese, S., & Eustice, R. (2012). Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. *Twenty-Sixth AAAI Conference on Artificial Intelligence*. https://doi.org/10.1609/aaai.v26i1.8379

Philion, J., & Fidler, S. (2020). Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16, Springer, (pp. 194–210).

Qi, C. R., Su, H., Mo, K., Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 652–660)

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems,30*.

Qi, C. R., Litany, O., He, K., & Guibas, L. J. (2019). Deep hough voting for 3d object detection in point clouds. *proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 9277–9286).

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., others (2021). Learning transferable visual models from natural language supervision. In: *International conference on machine learning*, PMLR, (pp. 8748–8763).

Reading, C., Harakeh, A., & Waslander, C. J.SL (2021) Categorical depth distribution network for monocular 3d object detection. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8555–8564).

Ren, S., Zeng, Y., Hou, J., & Chen, X. (2023). Corri2p: Deep image-to-point cloud registration via dense correspondence. *IEEE Transactions on Circuits and Systems for Video Technology, 33*(3), 1198–1208. https://doi.org/10.1109/tcsvt.2022.3208859

Schneider, N., Piewak, F., Stiller, C., & Franke, U. (2017). Regnet: Multimodal sensor registration using deep neural networks arXiv:1707.03167.

Sugimoto, S., Tateda, H., Takahashi, H., & Okutomi, M. (2004). Obstacle detection using millimeter-wave radar and its visualization on image sequence. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., (pp. 342–345) Vol. 3, https://doi.org/10.1109/ICPR.2004.1334537.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł, & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems,30*.

Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., & Novotny, D. (2025). Vggt: Visual geometry grounded transformer. *Proceedings of the Computer Vision and Pattern Recognition Conference* (pp. 5294–5306).

Wang, T., Zheng, N., Xin, J., & Ma, Z. (2011). Integrating millimeter wave radar with a monocular vision sensor for on-road obstacle detection applications. *Sensors, 11*(9), 8992–9008. https://doi.org/10.3390/s110908992

Wu, X., Jiang, L., Wang, P. S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., & Zhao, H. (2024). Point transformer v3: Simpler faster stronger. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 4840–4851).

Xiao, Y., Li, Y., Meng, C., Li, X., Ji, J., & Zhang, Y. (2024). Calibformer: A transformer-based automatic lidar-camera calibration network. arXiv:2311.15241.

Yang, C., Chen, Y., Tian, H., Tao, C., Zhu, X., Zhang, Z., Huang, G., Li, H., Qiao, Y., Lu, L., others (2023). Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (pp. 17830–17839).

Yin, T., Zhou, X., & Krahenbuhl, P. (2021). Center-based 3d object detection and tracking. In: *2021 IEEE/CVF Conference on Com-*

puter Vision and Pattern Recognition (CVPR). IEEE, https://doi.org/10.1109/cvpr46437.2021.01161.

Yuan, C., Liu, X., Hong, X., & Zhang, F. (2021). Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments. *IEEE Robotics and Automation Letters, 6*(4), 7517–7524.

Yuan, C., Liu, X., Hong, X., & Zhang, F. (2021). Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments. *IEEE Robotics and Automation Letters, 6*(4), 7517–7524.

Yuan, W., Li, J., Yue, J., Shah, D., Karydis, K., & Qiu, H. (2025). Bevcalib: Lidar-camera calibration via geometry-guided bird's-eye view representations arXiv:2506.02587.

Zhou, J., Ma, B., Zhang, W., Fang, Y., Liu, Y.-S., & Han, Z. (2023). Differentiable registration of images and lidar point clouds with voxelpoint-to-pixel matching. *Advances in Neural Information Processing Systems (NeurIPS),36*, 51166–51177.