Supplementary Material for "CalibBEV: LiDAR-Camera Calibration via BEV Alignment"

Filippo D'Addeo [*] [†]
filippo.daddeo2@unibo.it

Lorenzo Cipelli [*] [‡]
lorenzo.cipelli@unipr.it

Adriano Cardace [§]
cardace@stanford.edu

Emanuele Ghelfi [¶]
eghelfi@ambarella.com

Andrea Zinelli [¶]
azinelli@ambarella.com

Massimo Bertozzi [‡]
massimo.bertozzi@unipr.it

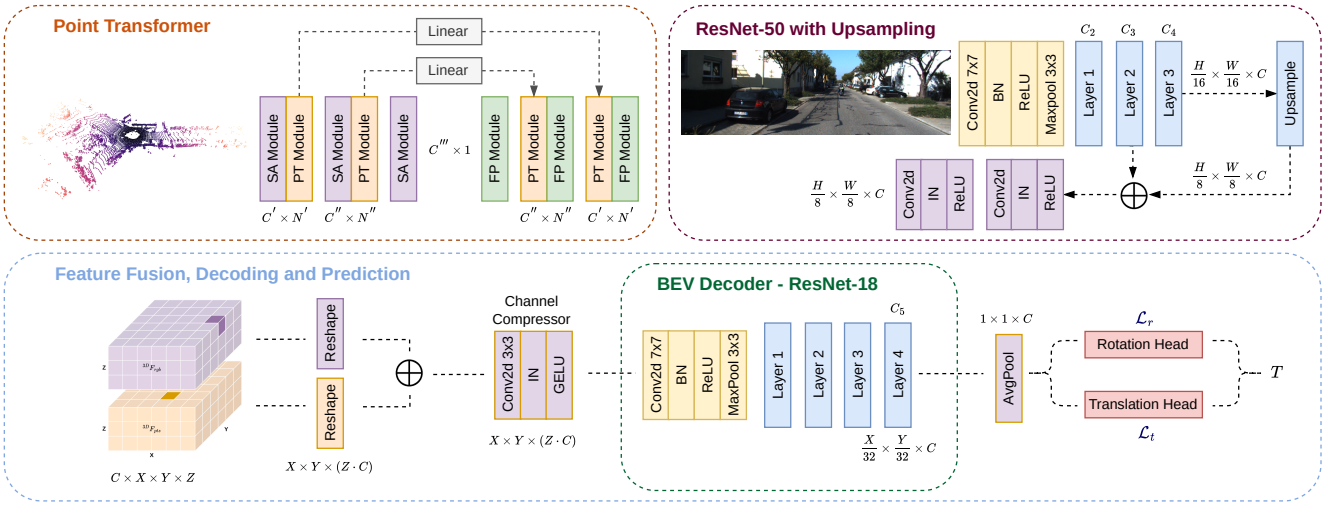


Figure 1. Images and point clouds are processed to extract two different "bird's eye view" (BEVs) representations of the same 3D space surrounding the vehicle. The two BEVs are then fused by concatenation and processed by a final decoder to predict the calibration matrix between the two sensors.

1. Network details

In this section, we dive into the details of `CalibBEV` architecture, first analyzing the modality-specific backbones, followed by an in-depth look at how the two-step alignment algorithm works with both textual and graphical descriptions.

1.1. Backbones

The point cloud backbone is inspired by [16], resembling the architecture of PointNet++ [12] in an encoder-decoder fashion, with Point Transformer blocks [15] in order to facilitate information exchange between these localized feature vectors, as shown in Fig. 1 (left). As regard as the RGB-specific backbone, we depict in Fig. 1 (right) the upsampling mechanism of the RGB features.

1.2. BEV Decoder

At the end of both the implicit and explicit alignments, a decoder module is implemented for decoding the fused BEVs features and reducing the channels dimension, enabling the final prediction to the two prediction heads. Inspired by [4], as shown in Fig. 1 we implement the BEV decoder as a ResNet-18 [5] to compute features maps of dimensions $X/32 \times Y/32 \times C$, which are reduced to features of dimensions $1 \times 1 \times C$ through a global average pooling.

[*] Equal contribution.
[†] University of Bologna, Department of Industrial Engineering, Italy.
[‡] University of Parma, Department of Engineering and Architecture, Italy.
[§] Stanford University, USA.
[¶] VisLab srl, an Ambarella Inc. company, Italy.

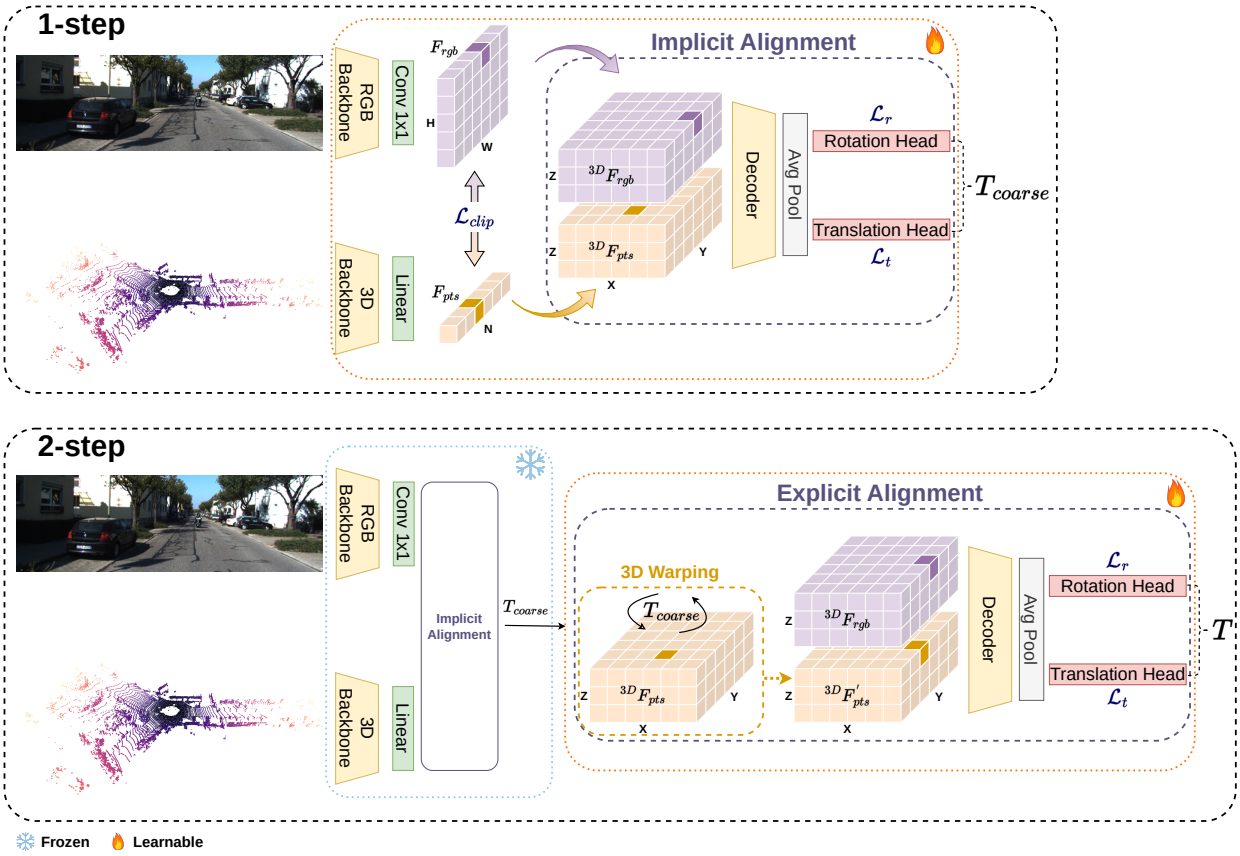


Figure 2. **BEV alignment.** We frame the LiDAR-camera calibration problem as a BEV alignment task. Domain-specific features are computed independently and projected into 3D BEV representations. Then, our BEV alignment module estimates the calibration matrix between the two sensors.

Algorithm 1 Training: first step

for $i \leftarrow 0$ to N **do** ▷ N number of iterations
  $I, P, T_{GT} \leftarrow$ Dataset[i]
  $F_{rgb} \leftarrow$ Conv1x1(RGB_Backbone(I))
  $F_{pts} \leftarrow$ Linear(3D_Backbone(P))
  $T_{coarse} \leftarrow$ Implicit_Alignment(F_{rgb}, F_{pts})
  $\mathcal{L} \leftarrow \alpha\mathcal{L}_{clip}(F_{rgb}, F_{pts}, T_{GT}) + \mathcal{L}_r(T_{coarse}, T_{GT})$
  $+\mathcal{L}_t(T_{coarse}, T_{GT})$

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

 # Save the Model trained weights
$W_{rgb} \leftarrow$ RGB_Backbone.save_weights()
$W_{pts} \leftarrow$ Point_Backbone.save_weights()
$W_{IA} \leftarrow$ Implicit_Alignment.save_weights()

1.3. Two-step Algorithm

Our two-step algorithm is designed to exploit both implicit and explicit alignment, leveraging the BEV representation of the RGB and 3D feature maps. Fig. 2 highlights the two-step training pipeline, in which we first train the two modality-specific backbones and the implicit alignment module (pseudo-code in Algorithm 1). Subsequently, the pre-trained weights are used to initialize the corresponding components in the second step and then all the learnable parameters from the previous step are frozen. We train the Explicit Alignment module with fewer iterations with respect to the first-step's training loop (pseudo-code in Algorithm 2) as the pre-initialization allows us to start from a better initialization point.

2. Additional Experiments and Analysis

2.1. RRE and RTE 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 [11]. The RRE

Method	KITTI (< 5m & $< 10°$)			nuScenes (< 5m & $< 10°$)		
	RTE(m)↓	RRE(°)↓	Acc.↑	RTE(m)↓	RRE(°)↓	Acc.↑
Grid Cls. + PnP [7]	1.07 ± 0.61	6.48 ± 1.66	49.67	2.35 ± 1.12	7.20 ± 1.65	58.90
DeepI2P (3D) [7]	1.27 ± 0.80	6.26 ± 2.29	51.46	2.00 ± 1.08	7.18 ± 1.92	17.02
DeepI2P(2D) [7]	1.46 ± 0.96	4.27 ± 2.74	59.98	2.19 ± 1.16	3.54 ± 2.51	83.50
CorrI2P [14]	0.74 ± 0.65	2.07 ± 1.64	89.06	1.83 ± 1.06	2.65 ± 1.93	89.30
VP2P-Match [16]	0.59 ± 0.57	2.39 ± 2.07	95.40	0.73 ± 0.65	1.39 ± 1.63	97.15
CurrI2P (VP2P-Match) [9]	0.44 ± 0.42	1.39 ± 1.44	95.98	-	-	-
CalibBEV (ours)	$\mathbf{0.04 \pm 0.10}$	$\mathbf{0.61 \pm 0.52}$	**99.96**	$\mathbf{0.04 \pm 0.02}$	$\mathbf{0.54 \pm 0.44}$	**100.0**

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. Results are filtered considering only samples with RTE $< 5m$ and RRE $< 10°$. '-' due to missing publicly available code.

Algorithm 2 Training: second step

 # Load the Model trained weights
RGB_Backbone.weights $\leftarrow W_{rgb}$
Point_Backbone.weights $\leftarrow W_{pts}$
Implicit_Alignment.weights $\leftarrow W_{IA}$
Explicit_Alignment.weights $\leftarrow W_{IA}$

 # Freeze modality-specific backbones
 # and implicit alignment module
RGB_Backbone.weights.no_grad()
Point_Backbone.weights.no_grad()
Implicit_Alignment.weights.no_grad()

for $j \leftarrow 0$ to $N^{'}$ **do** $\triangleright\ N^{'} << N$
  $I, P, T_{GT} \leftarrow$ Dataset[j]
  $F_{rgb} \leftarrow$ Conv1x1(RGB_Backbone(I))
  $F_{pts} \leftarrow$ Linear(3D_Backbone(P))
  $T_{coarse} \leftarrow$ Implicit_Alignment(F_{rgb}, F_{pts})
  $T \leftarrow$ Explicit_Alignment(F_{rgb}, F_{pts}, T_{coarse})
  $\mathcal{L} \leftarrow^{fine} \mathcal{L}_r(T_{coarse}, T, T_{GT})$
  $+^{fine} \mathcal{L}_t(T_{coarse}, T, T_{GT})$

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

corresponds to the summation of the absolute differences between the ground truth Euler angles and the predicted Euler angles:

$$\begin{cases} \text{RRE}_\theta = |\theta_{GT} - \theta_{pred}|, \\ \text{RRE}_\phi = |\phi_{GT} - \phi_{pred}|, \\ \text{RRE}_\psi = |\psi_{GT} - \psi_{pred}|, \\ \text{RRE} = \text{RRE}_\theta + \text{RRE}_\phi + \text{RRE}_\psi, \end{cases} \tag{1}$$

where ψ, ϕ and θ are respectively yaw, pitch, and roll. The RTE represents the magnitude of the difference between the ground-truth translation t_{GT} and the predicted translation t_{pred}:

Method	Size	KITTI		
		RTE(m)↓	RRE(°)↓	Acc.↑
VP2P-Match [16]	40×128	0.92	3.54	82.50
CalibBEV (ours)	40×128	**0.08**	**1.22**	**98.92**
VP2P-Match [16]	80×256	0.84	3.65	82.67
CalibBEV (ours)	80×256	**0.06**	**1.19**	**99.4**
VP2P-Match [16]	160×512	0.75	3.29	83.04
CalibBEV (ours)	160×512	**0.05**	**1.17**	**99.33**

Table 2. Results with different input RGB resolutions. Our Implicit Alignment module is more effective than VP2P-Match across different image resolutions. Values are from the original paper.

$$\text{RTE} = \|t_{GT} - t_{pred}\|_2. \tag{2}$$

2.2. Large Error Removal Testing

In order to avoid large errors due to failed registration, previous works, such as CorrI2P [14] and DeepI2P [7], compute the average RTE and RRE only for those samples with RTE lower than 5m and RRE lower than $10°$. Comparison against previous state-of-the-art methods in this specific setting is shown in Tab. 1. Once again, our model outperforms all the state-of-the-art methods by a significant margin, while yielding robuster predictions. We report once again the accuracy metric for completeness, although we argue that it is not a significant evaluation metric in the scenario described above.

2.3. Error Analysis

In this section, we evaluate the performance of CalibBEV against VP2P-Match across different mis-registration ranges in Fig. 3, 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 reaching 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

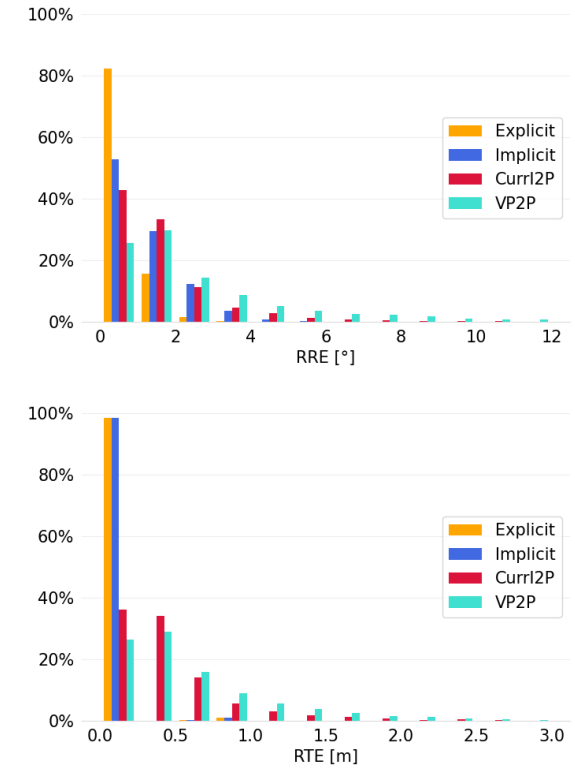


Figure 3. Histograms of image-point cloud registration RRE and RTE on the KITTI dataset. x-axis is RRE (°) and RTE (m), y-axis is the percentage.

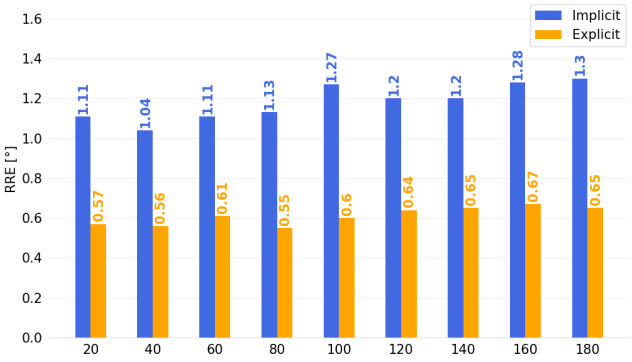


Figure 4. Rotation error for different initial mis-calibration on the KITTI dataset. x-axis is the initial mis-calibration interval in degrees, y-axis is the RRE (°) computed from all the samples that falls into the specific interval.

calibration errors, which is an essential property for online applications. The proposed Explicit Alignment is a step further to enhance robustness, yielding a more compact error distribution. To further demonstrate the effectiveness of our Explicit Alignment module, we analyze the mean error for different initial mis-registration intervals. As shown in Fig. 4 we gain a great advantage from the Explicit Alignment module in terms of RRE, as the new decoder is facilitated by the explicit alignment between BEVs features.

α	KITTI RTE(m)↓	KITTI RRE(°)↓	KITTI Acc.↑
1.0	0.07 ± 0.13	1.82 ± 2.70	95.59
0.5	$\mathbf{0.05 \pm 0.10}$	$\mathbf{1.17 \pm 1.56}$	**99.39**

Table 3. CLIP-based α hyperparameter analysis.

Method	KITTI RTE(m)↓	KITTI RRE(°)↓	KITTI Acc.↑
CorrI2P [14]	0.79 ± 3.27	2.36 ± 5.44	90.35
Calibnet [6]	0.29 ± 0.14	5.79 ± 2.43	47.69
LCCNet [10]	0.08 ± 0.04	0.89 ± 0.54	99.91
CalibBEV (ours)	$\mathbf{0.05 \pm 0.03}$	$\mathbf{0.50 \pm 0.28}$	**100.0**

Table 4. 6DoF registration analysis. For each axis sample a rotation between $\pm 5°$ and a translation between ± 0.5m.

2.4. CLIP-based Loss Hyperparameter Analysis

In Tab. 3 we compare different α weighting factors for the CLIP loss $\mathcal{L}_{clip}$ in our Implicit Alignment module. We found an optimal setup by setting this hyperparameter to 0.5. Notably, this value serves as an effective loss balancing factor, facilitating the training convergence.

2.5. Additional 6DoF Registration Analysis

Although this is not the standard benchmark neither for point-based methods nor for projection-based methods, in Tab. 4 we show the CalibBEV registration performances in a Six Degrees of Freedom (6DoF) scenario, where reduced variations for the three spatial coordinates and rotation angles are involved. Specifically, for each axis we sample both a random rotation and translation in range $\pm 5°$ and ± 0.5m, respectively. CalibBEV outperforms the latest state-of-the-art point-based approach with publicly available training code CorrI2P by 0.79m, 1.86°, and 9.65% on the RTE, RRE, and registration accuracy, respectively, also demonstrating the CalibBEV ability to estimate the height by encoding it into the channels dimension. For a fair and complete analysis, we also compare CalibBEV against Calibnet [6] and LCCNet [10], two state-of-the-art projection-based approaches. We outperform Calibnet by 0.24m, 5.29°, and 52.31% on the RTE, RRE, and registration accuracy, respectively, while we surpass LCCNet by 0.03m, 0.39°, and 0.09% on the RTE, RRE, and registration accuracy, respectively. Once again, CalibBEV shows to be more robust compared to all previous works that have been tested.

Method	nuScenes		
	RTE(m)↓	RRE(°)↓	Acc.↑
VP2P-Match [9]	0.89 ± 1.44	2.15 ± 7.03	88.33
CurrI2P(VP2P-Match) [9]	1.04 ± 1.64	2.67 ± 8.61	-
ICLM [8]	0.63 ± 0.44	2.13 ± 3.75	90.94
GraphI2P [1]	0.49 ± 1.22	1.73 ± 1.63	99.48
CalibBEV (ours) w/o acc.	0.03 ± 0.11	0.58 ± 2.17	99.90
CalibBEV (ours)	0.04 ± 0.08	0.54 ± 0.45	99.98

Table 5. Registration results against latest state-of-the-art approaches using point cloud accumulation on the nuScenes dataset.

2.6. Input Resolution Analysis

In Tab. 2 we explore how different image resolutions affect the registration performances. Specifically, we show that CalibBEV outperforms VP2P-Math in all the tested RGB input configuration settings when using only our Implicit Alignment-only model. Specifically, in the first two rows of Tab. 2 we set to 40×128 the input RGB dimensions. These are obtained by removing the top-50 rows, applying a 0.25 downsample factor, and randomly cropping the resulting images to match the desired dimensions. In this setting, our model outperforms VP2P-Match by 0.84m, by 2.32°, and by 16.42% in RTE, RRE, and registration accuracy, respectively. In the third and fourth rows of Tab. 2 we set the input RGB dimensions to be 80×256, with the same pre-processing stated above but with the downsample factor set to 0.125. Similarly, we outperform VP2P-Match by 0.78m, by 2.46°, and by 16.73% in RTE, RRE, and registration accuracy, respectively. Finally, in the last two rows where we use the input resolution also described in the main manuscript, we obtain the best performance overall, setting the new state-of-the-art for the benchmark.

2.7. RGB Backbone Analysis

Tab. 6 shows CalibBEV registration performances against different RGB backbones. Specifically, CalibBEV also outperforms VP2P-Match even with the same RGB backbone, while being comparable to the implementation described in the main manuscript.

2.8. Point Cloud Accumulation Analysis

In the main manuscript we stated that the point cloud accumulation is not strictly required for making CalibBEV work better than previous works. Indeed, in this section, differently from [1, 8, 9, 16] which set the cloud dimension to $N = 40960$, we downsample the point cloud dimension to $N = 36000$ as a single LiDAR sweep in nuScenes contains roughly 36k points. This should be a required step as differently from previous works, in this section we do not apply any accumulation technique on past and future frames. Indeed, we believe that accumulation from future frames hinders the applicability of the model in real-world scenarios, and accumulation from previous frames requires poses to be available. Therefore, Tab. 5 shows that CalibBEV achieves better registration performances than any other previous state-of-the-art method, even without accumulating the point cloud.

Method	Backbone	KITTI		
		RTE(m)↓	RRE(°)↓	Acc.↑
VP2P-Match [16]	Res-34	0.75 ± 1.13	3.29 ± 7.99	83.04
CalibBEV (ours)	Res-34	**0.03 ± 0.03**	0.70 ± 0.58	99.93
CalibBEV (ours)	Res-50	0.04 ± 0.10	**0.61 ± 0.52**	**99.96**

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

Method	Explicit	Time(s)↓
VP2P-Match [16]	-	0.2862
CurrI2P (VP2P-Match) [9]	-	0.2503
CalibBEV (ours)		0.1486
CalibBEV (ours)	✓	0.1695

Table 7. Inference time (s) on a NVIDIA RTX A6000 for each method with publicly available code. Explicit: Explicit Alignment module.

2.9. Inference Analysis

In Tab. 7, we report the inference time of CalibBEV compared to the latest works with a publicly available code, measured on an NVIDIA RTX A6000 GPU with a fixed batch size of 1. Our Implicit Alignment module is 0.10 and 0.13 faster than CurrI2P and VP2P-Match, respectively, while the Explicit Alignment runs 0.08 and 0.11 faster than CurrI2P(VP2P-Match), respectively. Overall, our model achieves a 32% reduction in inference time compared to CurrI2P.

3. Qualitative Analysis

3.1. CLIP Loss Effect on 2D-3D Features Similarity.

In Fig. 5, we present an additional display of images that further emphasize the significance of the CLIP-based [13] loss in measuring the similarity between 2D and 3D features. This visualization aims to provide a clearer understanding of how the CLIP-based loss plays a pivotal role in aligning these two distinct modalities, showing good results even on complicated associations. For example, in the first row of images we show how similar the features of humans are, respectively, represented by pixel features and point features. The second row displays the features similarity between road signs pixel features and point features.

3.2. Qualitative Self-Assessment Analysis

We conduct a qualitative self-assessment analysis both on the KITTI Odometry [3] and the nuScenes [2] datasets

to better understand the CalibBEV behavior. Indeed, in Figs. 6a, 6b, 7a and 7b 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. 6a shows two samples from the KITTI dataset in which CalibBEV achieves excellent results, scoring less than half-degree rotation error and a 0.05m translation error. Whereas, Fig. 6b shows two different circumstances in which CalibBEV outputs an inaccurate registration matrix, highlighting non-negligible rotation errors of $2.08°$ and $2.37°$, respectively.

Similarly, Fig. 7a shows some samples in which CalibBEV exhibits almost perfect registration performance on the nuScenes dataset, achieving a rotation error of $0.13°$ and $0.17°$, respectively, while highlighting a translation error of 0.03m and 0.02m respectively. However, for a fair and complete analysis on the nuScenes dataset, Fig. 7b shows two samples in which CalibBEV highlights non-negligible rotation errors in the predictions, with an RRE of $2.08°$ and $0.87°$.

3.3. Adverse Conditions Qualitative Self-Assessment Analysis

We also conduct a qualitative self-assessment on nuScenes under adverse weather conditions to better analyze potential CalibBEV limitations. In Fig. 8a we show two samples in which CalibBEV achieves excellent results, scoring a rotation error of $0.31°$ and $0.02°$, respectively. However, Fig. 8b shows two samples in which CalibBEV highlights non-negligible rotation errors, with an RRE of $1.27°$ and $1.45°$. 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 proposed multi-camera approach can provide richer RGB information, thereby enhancing registration performance.

References

[1] Lin Bie, Shouan Pan, Siqi Li, Yining Zhao, and Yue Gao. Graphi2p: Image-to-point cloud registration with exploring pattern of correspondence via graph learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22161–22171, 2025. 5

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 5

[3] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 5

[4] Adam W. Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. Simple-BEV: What really matters for multi-sensor bev perception? In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[6] Ganesh Iyer, R. Karnik Ram, J. Krishna Murthy, and K. Madhava Krishna. Calibnet: Geometrically supervised extrinsic calibration using 3d spatial transformer networks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018. 4

[7] Jiaxin Li and Gim Hee Lee. Deepi2p: Image-to-point cloud registration via deep classification. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 15955–15964. IEEE, 2021. 3

[8] Xinjun Li, Wenfei Yang, Jiacheng Deng, Zhixin Cheng, Xu Zhou, and Tianzhu Zhang. Implicit correspondence learning for image-to-point cloud registration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16922–16931, 2025. 5

[9] Liwei Lin, Chunyu Lin, Lang Nie, Shujuan Huang, and Yao Zhao. Curri2p: inter-and intra-modality similarity curriculum learning for image-to-point cloud registration. *The Visual Computer*, pages 1–14, 2025. 3, 5

[10] Xudong Lv, Boya Wang, Dong Ye, and Shuo Wang. Lccnet: Lidar and camera self-calibration using cost volume network, 2021. 4

[11] Yanxin Ma, Yulan Guo, Jian Zhao, Min Lu, Jun Zhang, and Jianwei Wan. Fast and accurate registration of structured point clouds with small overlaps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–9, 2016. 2

[12] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1

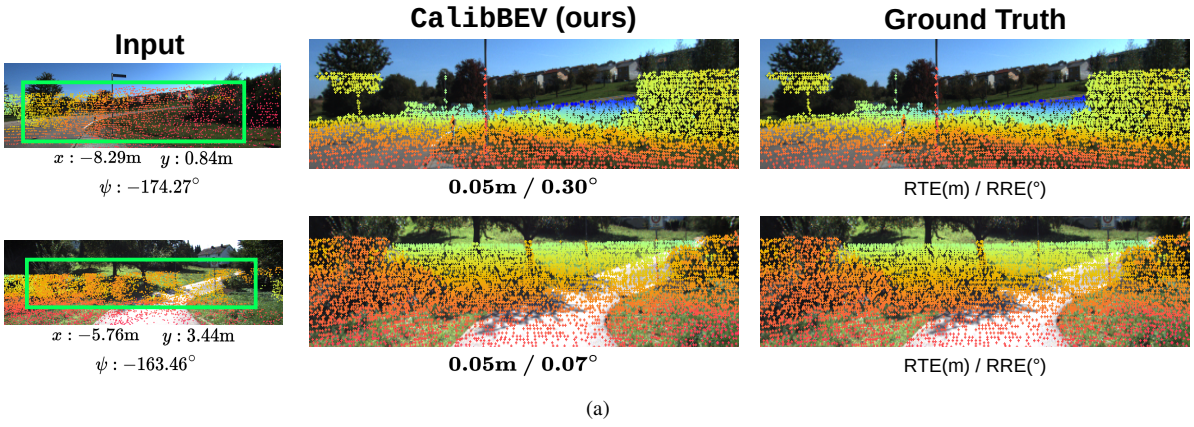
[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 5

[14] Siyu Ren, Yiming Zeng, Junhui Hou, and Xiaodong Chen. Corri2p: Deep image-to-point cloud registration via dense correspondence. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(3):1198–1208, 2023. 3, 4

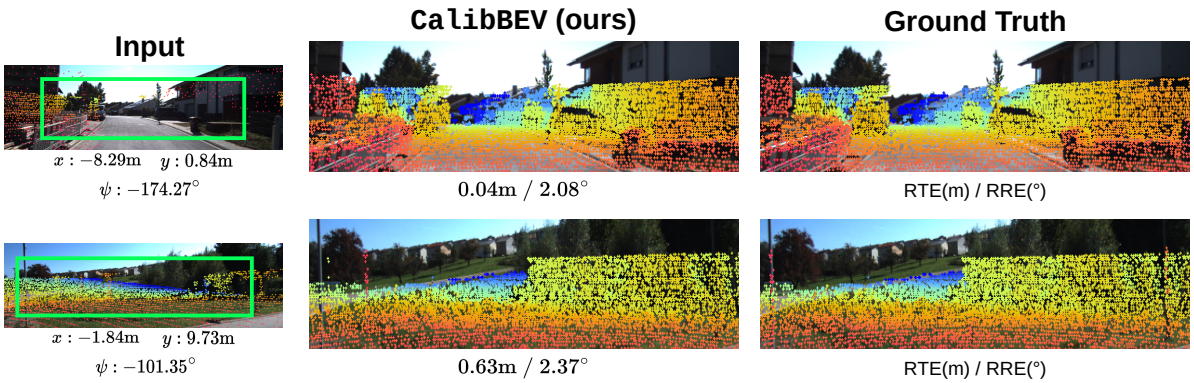
[15] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 1

[16] Junsheng Zhou, Baorui Ma, Wenyuan Zhang, Yi Fang, Yu-Shen Liu, and Zhizhong Han. Differentiable registration of images and lidar point clouds with voxelpoint-to-pixel matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1, 3, 5

Figure 5. **2D-3D feature similarity heat map.** For each couple of images, for a given 3D point projected into the image plane (bottom), we highlight the most similar pixels in features space (above). Semantic similarity across modalities facilitates the BEV alignment task. In the first row we highlight a strong similarity when picking a human, in the second row the subject of the similarity match is a road sign.



(a)



(b)

Figure 6. **KITTI qualitative analysis.** (a) Qualitative comparison of Image-to-Point Cloud registration results on the KITTI dataset. 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. (b) Examples of failure cases of Image-to-Point Cloud registration on the KITTI dataset. 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.

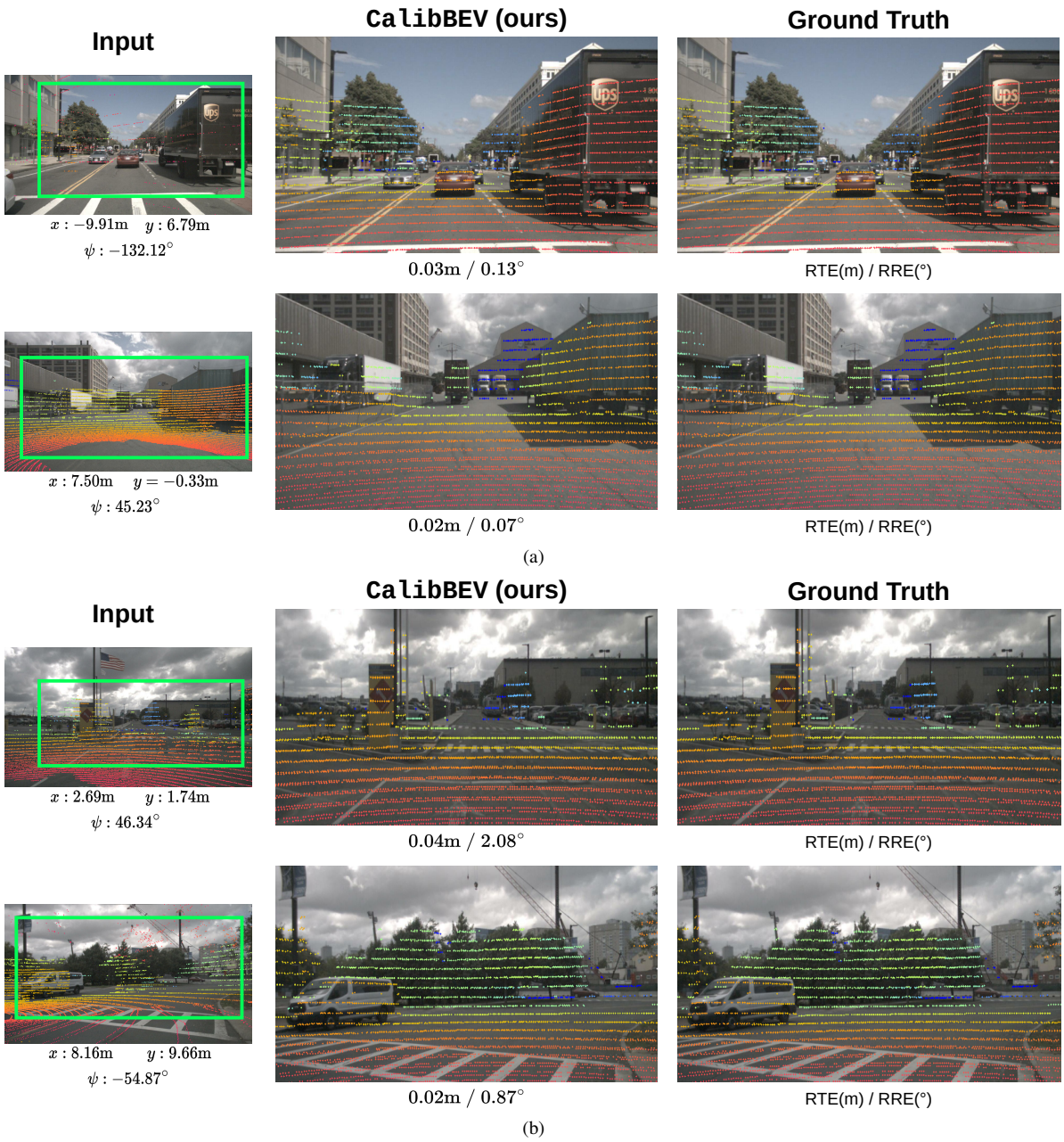


Figure 7. **nuScenes qualitative analysis.** (a) Qualitative comparison of Image-to-Point Cloud registration results on the nuScenes dataset. From left to right: mis-aligned point cloud and image inputs, our model calibration result, and the ground truth camera-LiDAR alignment. For each prediction, we report both RRE and RTE. (b) Examples of failure cases of Image-to-Point Cloud registration on the nuScenes dataset. 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.

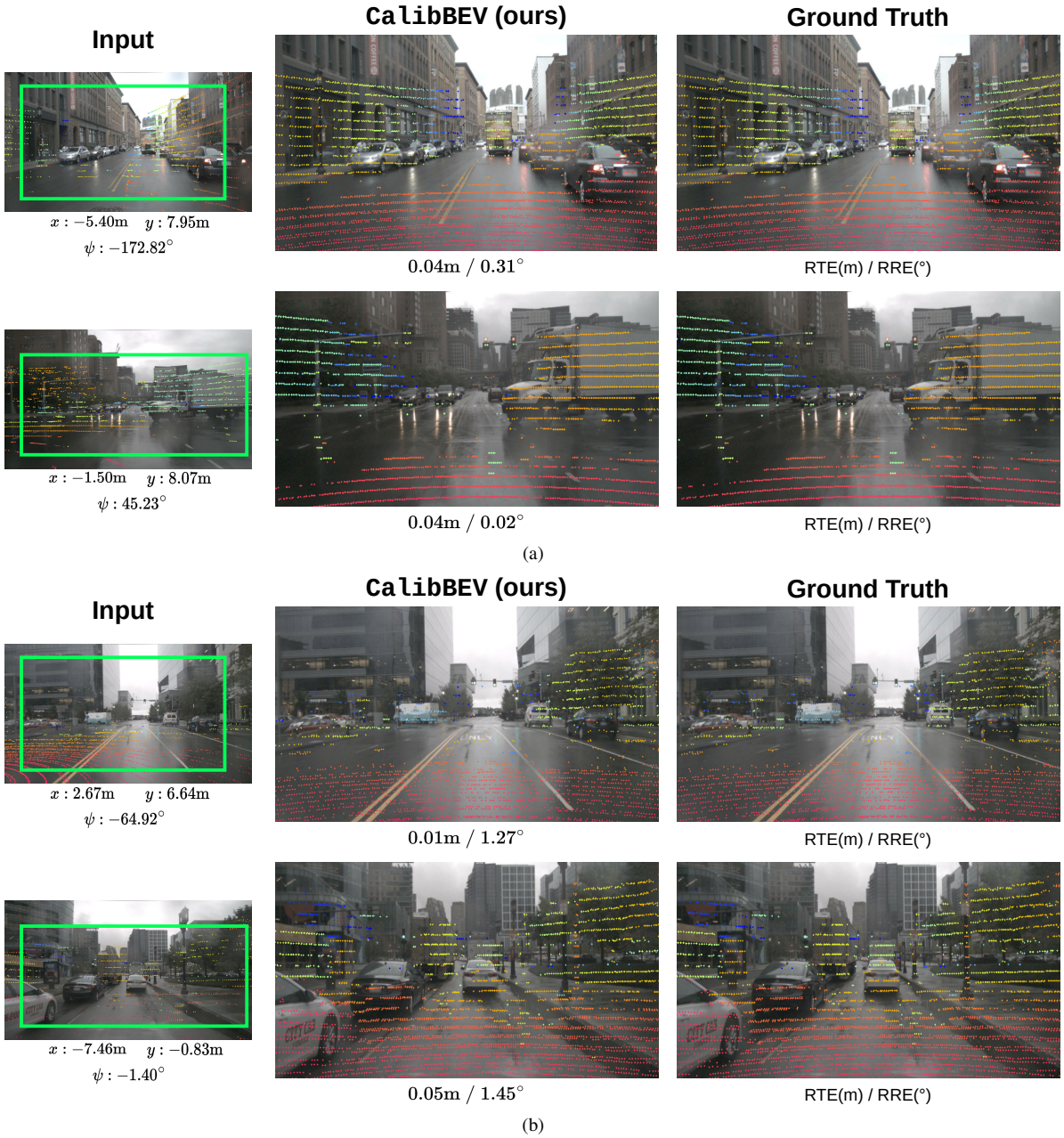


Figure 8. **nuScenes qualitative analysis in adverse weather conditions.** (a) Qualitative comparison of Image-to-Point Cloud registration results on the nuScenes dataset in rainy weather conditions. From left to right: mis-aligned point cloud and image inputs, our model calibration result, and the ground truth camera-LiDAR alignment. For each prediction, we report both RRE and RTE. (b) Examples of failure cases of Image-to-Point Cloud registration on the nuScenes dataset in rainy 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.