Trustworthy and Efficient Map-Free LiDAR Localization via Scan-Pose Alignment and Flow-Matching

Haoze Chang[1,2] **Dunqiang Liu**[1,2] **Minghang Zhu**[1,2] **Wen Li**[3]
Sheng Ao[1,2] **Siqi Shen**[1,2] **Chenglu Wen**[1,2] **Cheng Wang**[1,2*]
[1] Fujian Key Laboratory of Urban Intelligent Sensing and Computing, Xiamen University, Xiamen, China.
[2] Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, Xiamen, China.
[3] School of Engineering Mathematics and Technology, University of Bristol
hzchang@stu.xmu.edu.cn cwang@xmu.edu.cn

Abstract

Reliable map-free LiDAR relocalization requires not only accurate and efficient pose estimation but also trustworthy confidence estimation for detecting localization failures. However, estimating reliability to ensure trustworthy LiDAR relocalization remains largely unexplored. Also, existing LiDAR pose regression methods suffer from accuracy and efficiency trade-off. To address these limitations, we propose **SPAR** for confidence estimation and **VeLoc** for efficient pose regression. **SPAR** is a localization-method agnostic confidence estimator that reformulates localization reliability as scan-pose alignment. Given a query scan and a predicted pose, SPAR measures their compatibility in a learned embedding space and explicitly optimizes confidence scores to correlate scan with pose. **VeLoc** is proposed as an efficient and accurate pose regressor based on flow-matching for efficient pose refinement. Experiments on the Oxford and NCLT datasets demonstrate that the proposed framework achieves a favorable balance between efficiency and accuracy for LiDAR relocalization while enabling reliable localization failure detection across different localization backbones.

1 Introduction

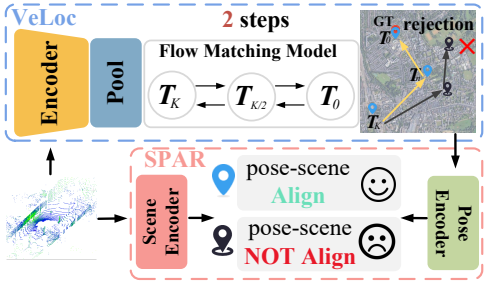


Figure 1: **Model overview.** Our method **VeLoc** achieves competitive performance compared to the SOTA baselines with faster speed. Also, our **SPAR** module effectively rejects outliers.

*Corresponding author.

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

LiDAR relocalization, defined as estimating 6-DoF pose from a LiDAR scan, is crucial for computer vision and robotics tasks [1–5]. Existing approaches can be categorized into map-based retrieval methods and map-free implicit methods [6]. Map-based methods maintain a scan database for retrieval [7] and result in burdensome map maintenance and storage overheads [8]. Consequently, implicit localization methods, which can be categorized into SCR [9, 10] and APR [11], are increasingly preferred [12].

Despite SCR achieving higher precision via correspondence estimation and RANSAC [13], it suffers from lower computational efficiency. Conversely, APR directly regresses poses, offering a simpler inference pipeline, lower runtime overhead, and better compatibility with other end-to-end tasks. However, current methods suffer from two limitations as follows.

First, current methods largely lack trustworthy evaluation of the reliability of pose predictions. We analyze the error distribution of leading methods like [9] and find out that pose estimations are highly accurate 95% of the time with errors under 4.8 meters. However, in the worst 1% of cases, the error drastically rises to over 11 meters, causing catastrophic deviations. Prior works mainly evaluate performance via average errors and lack such analysis. Such metrics inherently mask the rare but severe failures that are critical to safe autonomy. This issue necessitates an efficient uncertainty estimation method for outlier rejection. However, current uncertainty estimation approaches suffer from practical limitations including theoretical fragility and high computational costs [14–16]. Thus, how to design a localization-method agnostic confidence estimation method that can effectively reject outliers within only one forward pass becomes an important question.

Second, current methods fall short of achieving a favorable balance between accuracy and efficiency. As shown in Table 1, although DiffLoc and SGLoc achieve impressive accuracy, they suffer from slow inference speed. In contrast, restricting DiffLoc to two steps causes a dramatic performance drop, indicating that diffusion strongly relies on multi-step denoising, which inevitably weakens the efficiency advantage of APR. These observations motivate us to seek a few-step formulation that retains the benefit of refinement for accuracy while preserving a compact and efficient inference pipeline.

Table 1: **Accuracy-efficiency trade-off analysis.** We compare both APR and SCR methods on NCLT dataset.

Methods	Type	Trans. [m]	Rot. [°]	Latency [ms]
DiffLoc[17]	APR	✔**1.19**	✔**2.31**	✖**34.29**
DiffLoc(2-steps)[17]	APR	✖**8.68**	✖**7.00**	✔**9.44**
SGLoc[9]	SCR	✔**1.83**	✖**3.54**	✖**62.33**

Based on the observations above, we fully analyze the current localization and uncertainty estimation pipelines. For uncertainty estimation, inspired by VLM [18, 19], which explicitly learn a joint embedding space to measure cross-modal compatibility via symmetric contrastive learning, we reformulate confidence estimation as measuring the compatibility between the query scan and the predicted pose in a learned embedding space. Unlike heuristic methods that derive uncertainty without direct supervision or stochastic sampling from empirical statistics, our formulation yields an uncertainty estimation that is grounded in scan-pose correspondence. For efficiency and accuracy trade-off, we formulate iterative refinement in relocalization as progressively modifying the current pose estimate by modeling how the pose should gradually update toward its next state. The resulting accuracy-efficiency imbalance motivates us to model a few-step velocity field that better balances them. We therefore formulate iterative refinement as learning a conditional velocity field along a straight correction path, which aligns naturally with flow-matching.

In this paper, we propose SPAR, a plug-and-play uncertainty estimation module that enables robust failure detection, and VeLoc, a flow-matching-based APR framework that achieves efficient and accurate LiDAR relocalization, as shown in Figure 1. SPAR is designed as a dual-stream architecture with a queue of poses for negative samples. A sparse voxel-based GeoNet extracts geometric scan features, while multi-scale RFF encoders map poses into high-dimensional embeddings. Additionally, a novel compound loss and a coarse-to-fine strategy are designed for scan-pose alignment and score-error correlation. Furthermore, we propose VeLoc, introducing flow-matching into the pose regression. The input scan is projected into range images and processed using a ViT with DINO weights. Subsequently, a static-object-aware pool module (SOAP) [17]is deployed for moving objects removal. The contributions of our work are summarized as follows:

- First, we propose SPAR, a post-processing module based on embedding alignment to estimate the localization uncertainty. Our approach serves as a localization-method agnostic module that can effectively reject outliers after trained on relocalization scenes.
- Second, we propose VeLoc, a conditional flow-matching-based APR framework, which achieves competitive localization accuracy while reducing the localization inference latency by $3\times$ compared to the SOTA baselines.
- Third, we design a novel compound loss and a coarse-to-fine strategy tailored for scan-pose alignment. Experiments demonstrate that our design enforces a strong negative correlation between confidence scores and localization errors.

2 Related Work

2.1 Absolute Pose Regression and Generative Refinement

APR directly regresses the 6-DoF pose from the input in an end-to-end manner [20]. Beyond [20] for image localization, some image localization works focus on estimating the uncertainty of the regression results[15, 21–23] simultaneously. Based on [20], PointLoc [24] extends the pipeline to point clouds, employing a PointNet-style backbone[25] to handle unstructured data. Beyond PointLoc, various works have been proposed to improve regression accuracy [11, 24, 26–31]. Subsequent approaches enhance pose regression by incorporating auxiliary mechanisms, such as feature fusion [31], neuro-inspired memory [30], and spatio-temporal constraints [32]. Meanwhile, LiDAR-based odometry such as [33–35] also shares a similar regression pipeline and thus provides helpful insights for APR. Recently, generative models have been explored as refinement mechanisms for regression tasks. Diffusion model was initially proposed as a powerful generative model [36–39]. Grounded in Langevin dynamics, diffusion model effectively mitigates the training instability in GANs[40]. Recent works explore leveraging diffusion models as regressors for discriminative tasks [41, 42]. However, the inherent multiple denoising steps lead to slow inference speed[43, 44]. To address this bottleneck, flow-matching models[45, 46] have emerged as a promising direction, enabling efficient inference by modeling velocity to construct straight-line trajectories [47, 48].

2.2 Uncertainty Estimation

Uncertainty estimation has received increasing attention but remains largely underexplored in LiDAR-based relocalization. Existing image-based methods estimate uncertainty with auxiliary branches [14, 49] or predefined pose-error distributions [15, 22, 50]. Many of these estimates are empirically designed and lack strong theoretical grounding [16]. Generic techniques such as Monte Carlo (MC) dropout and deep ensembles are impractical for deployment, as they require multiple forward passes, and ensembles further require training multiple models. In LiDAR-based relocalization, DiffLoc [17] first uses sample variance as an uncertainty metric. However, variance can be poorly calibrated under tightly clustered samples and requires multiple predictions at inference time. Another branch of work estimates reliability from retrieval confidence. For example, HR-APR [51] uses the similarity score of the Top-1 retrieved result as an uncertainty indicator. This perspective suggests that uncertainty can be inferred from how well a query matches its reference representation. This naturally connects uncertainty estimation to contrastive learning, which has been widely used for cross-modal alignment [52, 53] and large-scale geolocalization [54, 55]. However, although conventional contrastive frameworks [56, 57] excel at discrete discrimination, they are inherently ill-suited for continuous regression. To bridge this gap, recent advances in continuous contrastive learning [58, 59] exploit global ranking to learn representations tailored to regression tasks.

3 Method

3.1 SPAR Module

To estimate the reliability of estimated pose, we propose **SPAR** (**S**can-**P**ose **A**lignment **R**eliability Estimator) module, which is designed as a dual-stream architecture to extract features from scene and pose, respectively, with a queue of past poses to expand the negative sample pool. The schematic overview is illustrated in Figure 2.

Scan encoder. The scan encoder consists of a scene feature branch and a geometric feature branch. Relying solely on range images (scene feature branch) is insufficient for distinguishing spatially adjacent scans due to their limited granularity and highly similar appearance. To address this limitation, we introduce a lightweight sparse voxel-based network GeoNet (geometric feature branch) to extract robust structural representations from point cloud voxels. The GeoNet adopts a U-Net structure with skip connections to fuse deep and shallow geometric features. Furthermore, we adopt a coordinate-aware pooling module at the output to preserve geometric details. Specifically, we explicitly inject the normalized coordinates $\mathbf{c} \in \mathbb{R}^3$ into the feature space before the final global pooling. This augmented tensor is then processed by an MLP to generate a position-aware feature embedding. Finally, we employ a gated fusion module which generates a soft mask to integrate the semantic information from the scene branch and the structural information from the geometry branch.

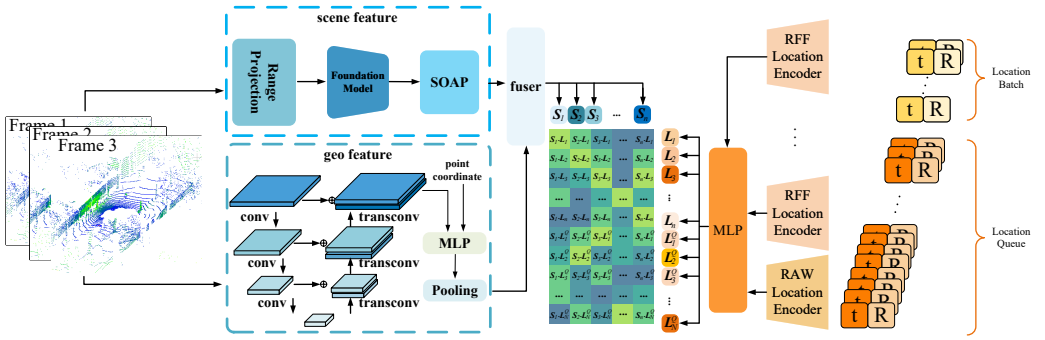


Figure 2: **SPAR module overall architecture.** The framework adopts a dual-stream design. The scan stream (blue) extracts and fuses scene feature and sparse voxel-based geometric feature. The pose stream (orange) encodes pose data using RFF and raw encoders. A pose memory queue stores past poses to expand the negative sample pool. Per-point features are fused with normalized coordinates via an MLP, followed by global max pooling.

Pose encoder. We use MLP architecture as pose encoder and find that standard MLPs typically lack the ability to capture high-frequency variations between different poses [60]. To mitigate this limitation, we leverage Random Fourier Features (RFF) [60] to map the input poses into higher-dimensional embeddings. Specifically, our encoder consists of N location capsules and a raw encoding capsule. For the i-th capsule, the input pose $\mathbf{p} \in \mathbb{R}^6$ is first mapped via Gaussian encoding with a specific frequency scale σ, followed by an MLP to extract parametric features. For σ, we employ a fixed interval schedule from 2^0 to 2^{10} to ensure discrimination for both global and local pose differences. Simultaneously, the raw capsule directly processes the original coordinates to preserve linear information. Finally, all features are concatenated and fused via a projection head. To learn a robust feature representation during training, a sufficiently large negative pool is a standard and essential component in contrastive learning [57]. However, the number of in-batch negatives is inherently constrained by the batch size, which limits sample diversity during training. To address this issue, we employ a pose queue $\mathcal{Q}$ to store past samples as additional negatives. At each forward pass, the queued poses are concatenated with the current in-batch samples to enlarge the negative pool. After each iteration, the queue is updated in a first-in-first-out (FIFO) manner, where newly generated negative samples are enqueued while the oldest samples are dequeued.

Training Pipeline SPAR employs a coarse-to-fine training strategy as outlined in Algorithm 1. For each training sample, S_i denotes the LiDAR scan, I_i its range image, and P_i the ground-truth pose. For a candidate pose $P_{i,j}$, we define its geometric error as $e_{i,j} = \mathrm{ADD}(P_{i,j}, P_i; S_i)$ evaluated in scaled coordinates. In each phase, a contrastive loss and a rank loss are employed for embedding alignment and for enforcing a negative correlation between scores and errors, respectively. We adopt an ADD-inspired formulation [61] to jointly account for rotational and translational discrepancies. See Appendix A for more details.

In the coarse inter-scan stage, each batch contains B samples with T frames per sample. We freeze the range-image encoder and randomly select one frame from each sample. The selected poses, together with historical poses in a FIFO queue, form the candidate pool for scan-pose alignment.

Random frame selection avoids populating the queue with densely spaced poses from consecutive frames. In the fine intra-scan stage, we perturb the ground-truth pose for the last frame in each sample to generate hard negatives and fuse range-image and voxel features. InfoNCE aligns each scan with its ground-truth pose, while the ranking loss orders the perturbed candidates according to their ADD errors.

The $\mathcal{L}_{\text{infonce}}^{\text{inter}}$ [57, 62] is employed to align scan and pose embeddings in the first phase. We design a novel loss $\mathcal{L}_{\text{rank}}^{\text{inter}}$ that can be formulated as $\mathcal{L}_{\text{rank}}^{\text{inter}} = 1 - \rho(\mathbf{S}_i^-, \mathbf{T}_i^-)$ where $\rho(\cdot,\cdot)$ denotes the Pearson coefficient. $\mathbf{S}_i^-$ contains the similarity scores of in-batch negative candidates, and $\mathbf{T}_i^-$ contains their ADD-based affinities $\exp(-e_{i,j}^2/(2\sigma_{\text{ADD}}^2))$, where σ_{ADD} is a bandwidth parameter. Queued poses are used only in the contrastive loss. Crucially, we exclude the ground truth (GT) sample from the ranking loss calculation to mitigate optimization conflicts in both phases. The second phase refines local discrimination by freezing low-frequency capsules and synthesizing noisy candidates. The $\mathcal{L}_{\text{infonce}}^{\text{intra}}$ can be formulated as:

$$\mathcal{L}_{\text{infonce}}^{\text{intra}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(s(I_i, P_{i,0}))}{\sum_{k=0}^{K-1}\exp(s(I_i, P_{i,k}))}, \tag{1}$$

where $P_{i,0} = P_i$ is the ground-truth pose and $P_{i,k}$, $k = 1, \ldots, K-1$, are perturbed candidates for the i-th scan.

Simultaneously, $\mathcal{L}_{\text{rank}}^{\text{intra}}$ imposes a pairwise ordering constraint as:

$$\mathcal{L}_{\text{rank}}^{\text{intra}} = -\frac{1}{|\mathcal{K}|}\sum_{(j,k)\in\mathcal{K}}\log\sigma\left(s_{i,j} - s_{i,k}\right), \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function. $\mathcal{K}$ contains ordered pairs of negative candidates (j, k) satisfying $e_{i,k} - e_{i,j} > \tau$, with $\tau = 0.05$. $s_{i,j}$ represents the similarity score $s(I_i, P_{i,j})$.

Algorithm 1 Training procedure for SPAR

Input: Pose $\mathbf{P}$, Scan $\mathbf{S} \in$ Dataset $\mathcal{D}$, Range Image $\mathbf{I}$ of scan $\mathbf{S}$, Encoders $\Phi_{\text{loc}}, \Phi_{\text{img}}, \Phi_{\text{geo}}$, Queue $\mathcal{Q}$, Similarity matrix $\mathbf{Z}$, Relative point-wise error $\mathbf{E}$
if Stage == 'inter-scan' **then**
  Stage 1: inter-scan training. Freeze Φ_{img} *freeze the range image encoder*
  for each batch $(\mathbf{P}, \mathbf{S}, \mathbf{I}) \in \mathcal{D}$ **do**
    Sample $t \sim \mathcal{U}[0, T-1]$; $\mathbf{f}_{\text{img}} \leftarrow \Phi_{\text{img}}(\mathbf{I}_t)$ *randomly sample a frame*
    $\mathbf{P}_{\text{set}} \leftarrow [\mathbf{P}_{\text{t}}; \mathcal{Q}]$; **Update** $\mathcal{Q}$; *enqueue current pose*
    $\mathbf{Z} \leftarrow \text{Sim}(\mathbf{f}_{\text{img}}, \Phi_{\text{loc}}(\mathbf{P}_{\text{set}}))$ *compute similarity matrix*
    $\mathbf{Z}^-, \mathbf{E}^- \leftarrow \text{InBatchNegatives}(\mathbf{Z}, \mathbf{P}_t, \mathbf{S}_t)$ *exclude GT and queued poses*
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{InfoNCE}}(\mathbf{Z}) + \lambda\mathcal{L}_{\text{Rank}}(\mathbf{Z}^-, \mathbf{E}^-)$; **Update** trainable SPAR parameters
  end for
else
  Stage 2: intra-scan training. Freeze Φ_{img} and low-freq capsules in Φ_{loc} *freeze low-freq components*
  for each batch $(\mathbf{P}, \mathbf{S}, \mathbf{I}) \in \mathcal{D}$ **do**
    $\mathbf{P}_{\text{neg}} \sim \mathcal{N}(\mathbf{P}_{\text{gt}}, \sigma)$; *Generate perturbed poses around GT*
    $\mathbf{f}_{\text{scan}} \leftarrow \text{Fuse}(\Phi_{\text{img}}(\mathbf{I}), \Phi_{\text{geo}}(\mathbf{S}))$ *fuse scan-level features*
    $\mathbf{s} \leftarrow \text{Sim}(\mathbf{f}_{\text{scan}}, \Phi_{\text{loc}}([\mathbf{P}_{\text{gt}}; \mathbf{P}_{\text{neg}}]))$; $\mathbf{E}_{\text{neg}} \leftarrow \text{ADD}(\mathbf{P}_{\text{neg}})$ *score perturbed poses*
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{InfoNCE}}(\mathbf{s}) + \mathcal{L}_{\text{Rank}}(\mathbf{s}, [\mathbf{0}; \mathbf{E}_{\text{neg}}])$; **Update** unfrozen parameters *ranking internally excludes GT*
  end for
end if

3.2 VeLoc

To realize accurate and efficient localization, we propose VeLoc, following the paradigm of APR methods which consists of a feature learner and a pose regressor as illustrated in Figure 3.

Feature representation. To leverage the structural regularity and computational efficiency of 2D-conv in unstructured point clouds, we project the point cloud $\mathcal{P} = \{(x_i, y_i, z_i)\}_{i=1}^{N}$ to range image grid of size $H \times W$. We sort the candidates by range depth and retain the point with the minimum depth when multiple points are projected to the same pixel. Then, we leverage the representation

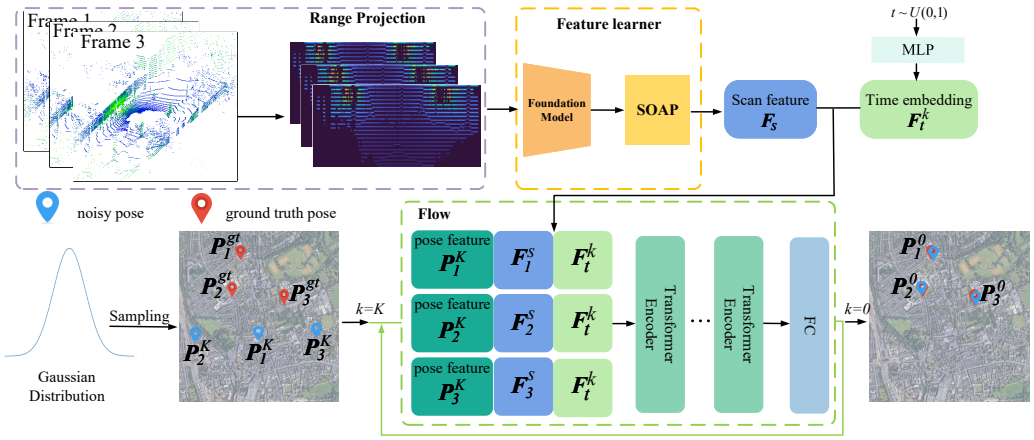


Figure 3: **VeLoc overall architecture.** Input LiDAR frames are first projected into range images. The feature learner module comprises a foundation model and a SOAP module. The pose regression task takes the scan features $\mathbf{F}_i^{\mathrm{s}}$, a shared time embedding $\mathbf{F}_t^{\mathrm{k}}$, and noisy poses $\mathbf{P}_i^K$ sampled from a Gaussian distribution as input. A Transformer-based flow network generates the final pose $\mathbf{P}_i^0$.

ability of foundation models trained on large-scale datasets following prior works [17, 63]. To bridge the domain gap between range and RGB images, we replace the embedding layer with convolution stem and feed the output to a ViT with DINO parameters. Finally, we incorporate the SOAP module in DiffLoc[17] to suppress artifacts from moving objects[64]. We represent the pose as $y = [t, r] \in \mathbb{R}^6$, where r is the axis-angle rotation obtained by applying the logarithm map to the quaternion. This avoids unit-norm constraints of quaternion and gimbal lock issues of Euler angles. Flow-matching is performed in the Euclidean $\mathbb{R}^6$ space without enforcing the $SO(3)$ geometry, providing an efficient approximation for autonomous driving scenarios. More details are shown in Appendix.A.3

flow-matching pose regressor. We formulate pose regression as a conditional flow-matching problem which constructs a deterministic optimal transport path from a standard Gaussian distribution $x_0 \sim \mathcal{N}(0, I)$ to the target pose x_1, which is defined as $x_t = (1-t)x_0 + tx_1$, parameterized by time $t \in [0, 1]$. This linear interpolation defines a straight-line velocity field and its time derivative corresponds to a constant velocity field derived as $u_t(x_t|x_1, x_0) = \frac{d}{dt}x_t = x_1 - x_0$. The network v_θ is trained to regress the target velocity by minimizing the L1 loss

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{t,x_0,x_1}\left[\|v_\theta(x_t, t, c) - (x_1 - x_0)\|_1\right]. \tag{3}$$

where c denotes the query scan embedding. During inference, given a query scan embedding c and sampled noise x_0, the final pose $\hat{x}_1$ is recovered by numerically integrating the velocity field from $t = 0$ to $t = 1$. In practice, we approximate this integral using numerical ordinary differential equation (ODE) solvers (e.g., Euler method[65]) with discrete time steps.

4 Experiments

4.1 Experiments Setup

We conduct experiments on the NCLT[66] and Oxford [67] datasets. Our method is implemented with MinkowskiEngine[68] and PyTorch. Further training details are provided in the Section A.

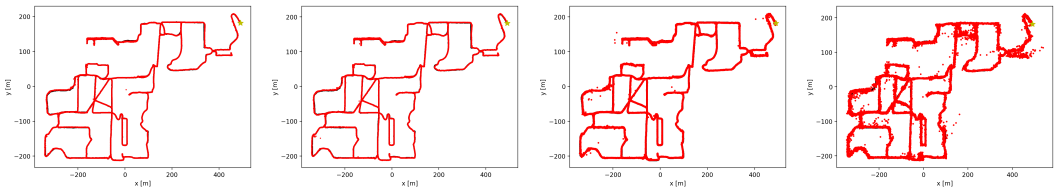
4.2 Localization Accuracy Comparison

NCLT dataset. Table 2 summarizes the results on the NCLT dataset. VeLoc achieves the best performance on both metrics against all evaluated baselines. First, VeLoc outperforms LightLoc, reducing translation and rotation error by 30.1% and 18.2%, respectively. Our generative paradigm effectively preserves localization consistency, whereas SCR methods often suffer from feature

mismatching in dynamic environments. Second, compared with DiffLoc, the strongest APR baseline, VeLoc achieves an approximately 3 $\times$ speedup while reducing the average translation and rotation errors. compared with HypLiLoc, the strongest discriminative APR baseline, VeLoc reduces the average translation and rotation errors by 47.7% and 27.5%, respectively. Visualizations are provided in Figure 4.

Table 2: **Comparison with the state-of-the-art methods on NCLT dataset.** The mechanism (Mech.) S/A denotes the structure-based localization/absolute pose regression. We highlight the best and second-best results.

Methods	Mech.	2012-02-12	2012-02-19	2012-03-31	2012-05-26	Average [m/°]	Ranks
SGLoc[9]	S	1.20/3.08	1.20/3.05	1.12/3.28	3.81/4.74	1.83/3.54	4/6
LightLoc[69]	S	0.98/2.76	0.89/2.51	0.86/2.67	3.10/3.26	1.46/2.80	3/3
PoseSOE[11]	A	13.09/8.05	6.16/4.51	5.24/4.56	12.60/7.67	9.27/6.20	8/8
STCLoc[32]	A	4.91/4.34	3.25/3.10	3.75/4.04	8.67/5.23	5.15/4.18	7/7
NIDALoc[30]	A	4.48/3.59	3.14/2.52	3.67/3.46	6.60/4.56	4.47/3.53	6/5
HypLiLoc[31]	A	1.71/3.56	1.68/2.69	1.52/2.90	2.90/3.47	1.95/3.16	5/4
DiffLoc[17]	A	0.99/2.40	0.92/2.14	0.98/2.27	1.88/2.43	1.19/2.31	2/2
VeLoc(Ours)	A	0.82/2.44	0.73/2.12	0.77/2.23	1.78/2.39	1.02/2.29	1/1



(a) Ours(0.73/2.12/**12.39**) (b) DiffLoc(0.92/2.14/34.29) (c) SGLoc(1.20/3.05/62.33) (d) NIDALoc(3.14/2.52/110.28)

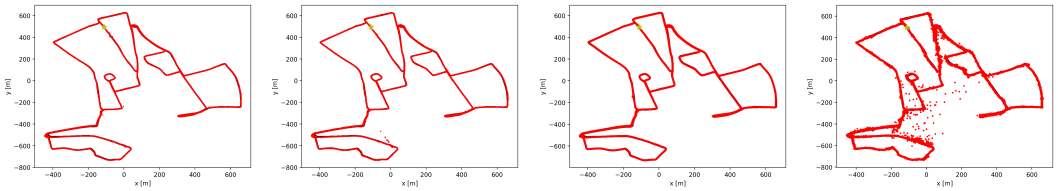
Figure 4: **Visualization of trajectories on the NCLT dataset.** The starting frame is marked by a star. The black and red lines represent the ground truth and the predicted paths, respectively. **Latency** is measured with a batch size of 1 to simulate real-world usage.

Oxford dataset. As detailed in Table 3 and Figure 5, our proposed VeLoc achieves competitive performance against evaluated baselines. First, compared to APR baselines, VeLoc significantly reduces localization errors. Specifically, VeLoc reduces the translation and rotation errors, with 2$\times$ speedup when compared to APR SOTA, DiffLoc. Second, in comparison to LightLoc, which ranks first in translation accuracy, VeLoc reduces the rotation error by nearly twofold. While SCR methods excel at position alignment, our regression approach offers superior robustness in estimating rotation. More analyses are shown in Appendix.B.1. More visualizations on both datasets are provided in Section B.7.

Table 3: **Comparison with the baseline methods on Oxford dataset.** The mechanism (Mech.) S/A denotes the structure-based localization/absolute pose regression. We highlight the best and second-best results.

Methods	Mech.	15-13-06-37	17-13-26-39	17-14-03-00	18-14-14-42	Average [m/°]	Ranks
SGLoc[9]	S	3.01/1.91	4.07/2.07	3.37/1.89	2.12/1.66	3.14/1.88	2/7
LightLoc[69]	S	2.33/1.21	3.19/1.34	3.11/1.24	2.05/1.20	2.67/1.25	1/4
PoseSOE[11]	A	7.59/1.94	10.39/2.08	9.21/2.12	7.27/1.87	8.62/2.00	8/8
STCLoc[32]	A	6.93/1.48	7.55/1.23	7.44/1.24	6.13/1.15	7.01/1.28	7/5
NIDALoc[30]	A	5.45/1.40	7.63/1.56	6.68/1.26	4.80/1.18	6.14/1.35	6/6
HypLiLoc[31]	A	6.88/1.09	6.79/1.29	5.82/0.97	3.45/0.84	5.74/1.05	5/3
DiffLoc[17]	A	3.57/0.88	3.65/0.68	4.03/0.70	2.86/0.60	3.53/0.72	4/2
VeLoc(Ours)	A	3.30/0.92	3.76/0.62	3.94/0.59	2.66/0.55	3.42/0.66	3/1

Localization Efficiency We further investigate the efficiency of VeLoc under different sampling steps in Table 4. VeLoc achieves strong few-step performance on both datasets. On Oxford, it reaches



(a) Ours(2.66/0.55/**14.06**) (b) DiffLoc(2.86/0.60/26.58) (c) SGLoc(2.12/1.66/90.12) (d) NIDALoc(4.80/1.18/120.23)

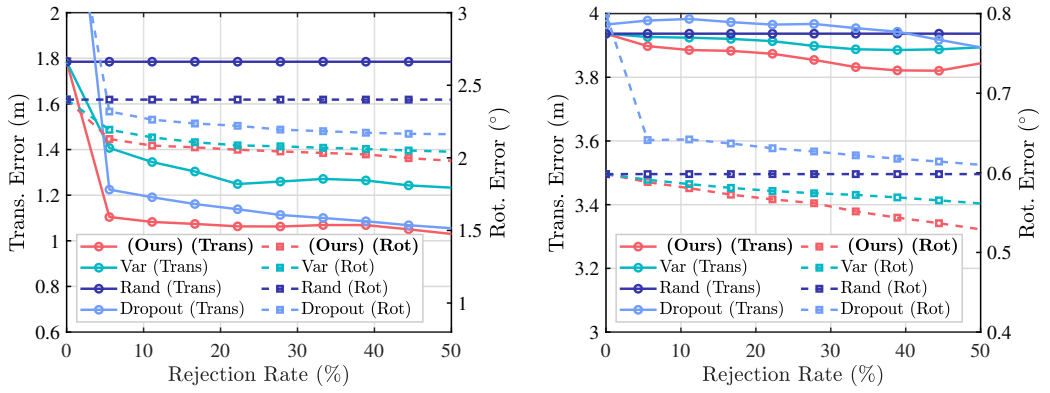
Figure 5: **Visualization of trajectories on the Oxford dataset.** The starting frame is marked by a star. The black and red lines represent the ground truth and the predicted paths, respectively. **Latency** is measured with a batch size of 1 to simulate real-world usage.

3.44m/0.71° with only 2 steps, outperforming DiffLoc even with 20 steps and providing about 4× fewer sampling steps for comparable accuracy. On NCLT, VeLoc achieves 1.02m/2.29° with few steps, while DiffLoc suffers severe degradation at 2 steps and still lags behind in translation accuracy at 20 steps. More results are shown in Appendix.B.2. Table 4 reports sampling time in milliseconds per batch of 28, rather than per-sample latency. Its purpose is to compare localization accuracy and runtime as the sampling-step count changes under the same conditions and on the same computing hardware. The actual latency claim for online usage should therefore be read from subplots below Figure 5 and Figure 4.

Table 4: **Impact of sampling steps.** Errors are reported as translation (m) / rotation (°), and latency denotes the sampling time (ms) per batch. Batch size is set to 28 during evaluation.

Steps	NCLT				Oxford			
	VeLoc(Ours)		DiffLoc		**VeLoc(Ours)**		DiffLoc	
	Error ↓	Latency ↓	Error ↓	Latency ↓	Error ↓	Latency ↓	Error ↓	Latency ↓
2	1.05 / 2.35	5.80	8.68 / 7.00	4.76	3.44 / 0.71	4.45	13.73 / 3.38	4.59
4	**1.02 / 2.29**	9.43	1.57 / 2.57	8.59	3.40 / 0.68	7.78	4.04 / 0.85	8.86
8	1.02 / 2.29	18.13	1.30 / 2.35	16.73	3.42 / 0.68	15.41	3.55 / 0.75	16.88
10	1.03 / 2.30	21.04	1.26 / 2.33	20.88	**3.42 / 0.66**	17.99	3.53 / 0.72	21.12
20	1.04 / 2.32	40.83	**1.19 / 2.31**	40.68	3.44 / 0.68	34.87	**3.49 / 0.72**	41.75

4.3 Uncertainty Estimation Evaluation



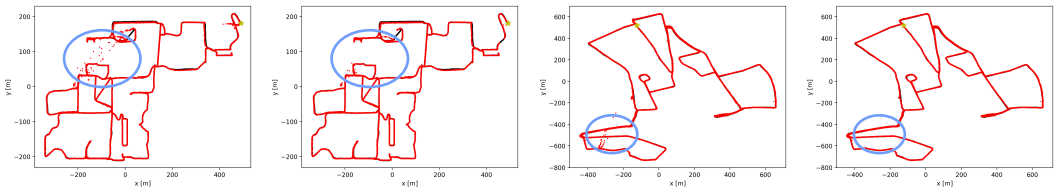
(a) NCLT. (b) Oxford.

Figure 6: **Error rejection curves on VeLoc.** The solid lines and dashed lines represent translation (left Y-axis) and rotation (right Y-axis) errors, respectively.

Failure detection for safe deployment. Detecting localization failures is indispensable for safe deployment. We evaluate failure detection capability using error-rejection curves by progressively excluding lowest confidence score samples from VeLoc and DiffLoc to validate the localization-method agnostic characteristic as shown in Figure 6. We also adopt variance, which is computed

from multiple samples under fixed model parameters, as well as MC dropout as baselines. However, we note here that dropout layer is not implemented in many deterministic architectures including VeLoc, rendering MC Dropout sampling inapplicable. We adopt a post-hoc dropout approximation by injecting a dropout layer at test-time following [70]. Appendix Section B.3 additionally reports failure-detection AUROC under a 10 m translation-error threshold and failure recall at a 5% rejection budget on selected heavy-tailed sequences.

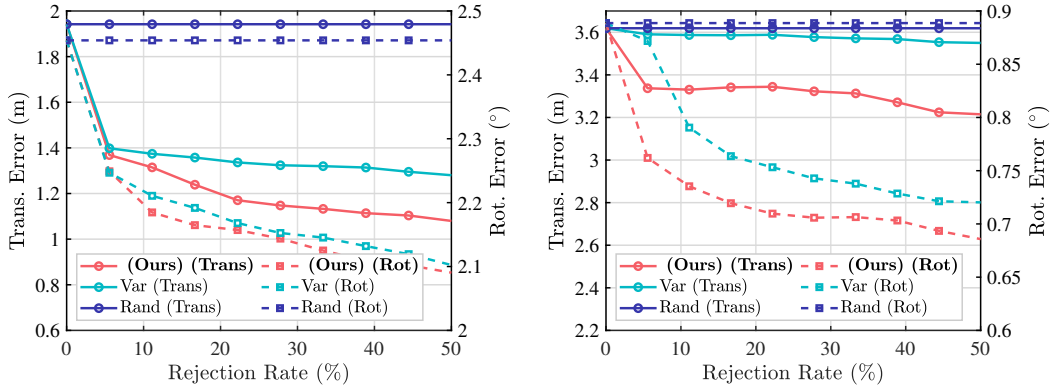
As illustrated in Figure 6, both translation and rotation errors decrease as the rejection rate increases. This trend validates that our SPAR effectively captures the uncertainty and serves as a reliable indicator of localization quality. Crucially, SPAR consistently outperforms the variance baseline across all rejection levels. Notably, the steepest error drop occurs within the first 5% of rejection on the test trajectory, which indicates that SPAR is highly efficient at filtering out outliers as shown in Figure 7. See Appendix B.3 for more results.



(a) w/o rejection. (b) w/ rejection. (c) w/o rejection. (d) w/ rejection.

Figure 7: **Visualization of trajectory after rejection**. The results illustrate that our method can effectively filter outliers. We reject the lowest 10% and 2% confident samples for Oxford and NCLT datasets, respectively. The blue circles highlight the significant outliers.

To further validate the generalizability and localization-method agnostic capabilities of our proposed SPAR module, we extend our evaluation to DiffLoc [17] as shown in Figure 8.

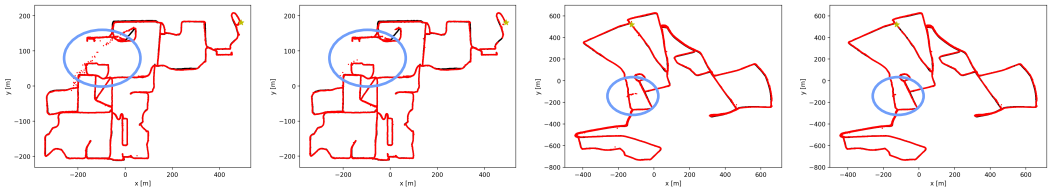


(a) NCLT. (b) Oxford.

Figure 8: **Error rejection curves on DiffLoc.** The solid lines and dashed lines represent translation (left Y-axis) and rotation (right Y-axis) errors, respectively.

As illustrated in the figure, on the NCLT dataset, localization error decreases progressively as the rejection rate increases. Moreover, the improvement magnitude achieved by the SPAR module surpasses that of the baseline using variance as an uncertainty indicator. However, on the Oxford dataset, although a similar overall trend is observed, the translation error does not show a significant decrease. This discrepancy is primarily attributed to the inherent complexity of the Oxford dataset, which presents more challenging scenarios.

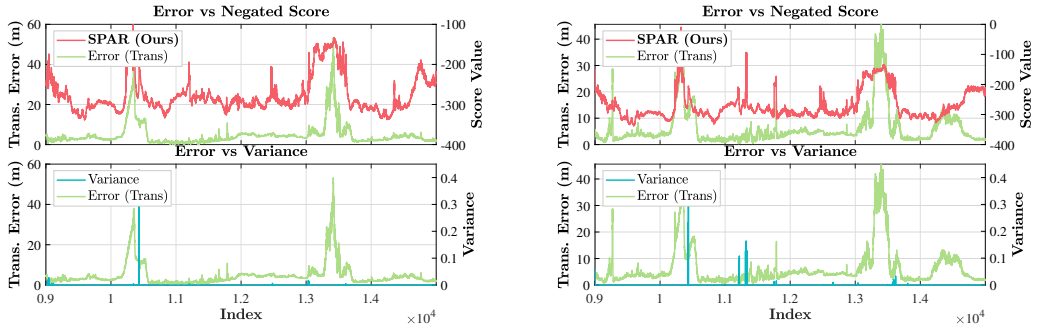
The estimated trajectories by filtering out the samples with the lowest confidence scores are shown in Figure 9. Visualizations of the trajectories confirm that SPAR effectively identifies and excludes geometric outliers.



(a) NCLT w/o rejection (b) NCLT w/ rejection (c) Oxford w/o rejection (d) Oxford w/ rejection

Figure 9: **Visualization of trajectory after rejection**. The results illustrate that our method can effectively filter outliers. We reject the lowest 10% and 2% confident samples for Oxford and NCLT datasets, respectively. The blue circles highlight the significant outliers.

Score-error alignment. Beyond outlier rejection, confidence scores should correlate negatively with pose error. However, very few works validate this property.



(a) Results on VeLoc. (b) Results on DiffLoc.

Figure 10: **Pose estimation uncertainty visualization**. We show the relationship between localization error (left y-axis) and variance alongside confidence score estimation result (right y-axis) on the Oxford and NCLT dataset. The confidence scores **are negated** to align with the trend of localization error and variance.

To illustrate, we select a representative segment to visualize the correlation between localization error and uncertainty metrics as shown in Figure 10, where confidence scores are negated to align with the error trend. The confidence scores from our SPAR module demonstrate strong negative correlation with translation errors. In contrast, the variance collapses towards zero due to the high determinism of the localization model and thus exhibits weak correlation, failing to scale with the actual error. All correlation coefficients are computed using the original, non-negated confidence scores. Our model yields the Pearson coefficient $r_p = -0.6856$ and Spearman's rank coefficient $r_s = -0.6040$ on NCLT dataset and $r_p = -0.6890$ and $r_s = -0.6161$ on Oxford dataset, indicating a negative correlation. Crucially, our method serves as a localization-method agnostic, plug-and-play solution that provides uncertainty estimation, which we validate with multiple localization methods. Further experiments are shown in Appendix B.4.

5 Conclusion

In this paper, we propose SPAR, a plug-and-play module that reformulates uncertainty estimation as an alignment task via contrastive learning. We further propose VeLoc, a flow-matching framework for LiDAR relocalization, achieving efficient and precise pose estimation. Experiments demonstrate that VeLoc accelerates by $3\times$ compared to the SOTA APR baseline while achieving superior accuracy. Crucially, SPAR exhibits robust and reliable failure detection across multiple localization methods. In summary, our framework achieves a favorable balance between efficiency and accuracy for LiDAR relocalization and realizes reliable failure detection across different localization backbones.

Acknowledgments and Disclosure of Funding

This work was partially supported by the Fundamental Research Funds for the Central Universities (20720260165) and the National Natural Science Foundation of China (No. 62501502). The authors declare no competing interests.

References

[1] Jingyuan Zhao, Yuyan Wu, Rui Deng, Susu Xu, Jinpeng Gao, and Andrew Burke. A survey of autonomous driving from a deep learning perspective. *ACM Computing Surveys*, 57(10):1–60, 2025.

[2] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12): 4338–4364, 2020.

[3] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *CVPR*, pages 11753–11762, 2021.

[4] Sheng Ao, Yulan Guo, Qingyong Hu, Bo Yang, Andrew Markham, and Zengping Chen. You only train once: Learning general and distinctive 3d local descriptors. *TPAMI*, 45(3):3949–3967, 2022.

[5] Sheng Ao, Qingyong Hu, Hanyun Wang, Kai Xu, and Yulan Guo. Buffer: Balancing accuracy, efficiency, and generalizability in point cloud registration. In *CVPR*, pages 1255–1264, 2023.

[6] Athanasios Chalvatzaras, Ioannis Pratikakis, and Angelos A Amanatiadis. A survey on map-based localization techniques for autonomous vehicles. *IEEE Transactions on intelligent vehicles*, 8(2):1574–1596, 2022.

[7] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4470–4479, 2018.

[8] Jacek Komorowski. Minkloc3d: Point cloud based large-scale place recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1790–1799, 2021.

[9] Wen Li, Shangshu Yu, Cheng Wang, Guosheng Hu, Siqi Shen, and Chenglu Wen. Sgloc: Scene geometry encoding for outdoor lidar localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9286–9295, 2023.

[10] Jianshi Wu, Minghang Zhu, Dunqiang Liu, Wen Li, Sheng Ao, Siqi Shen, Chenglu Wen, and Cheng Wang. Leader: Learning reliable local-to-global correspondences for lidar relocalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9932–9942, June 2026.

[11] Shangshu Yu, Cheng Wang, Chenglu Wen, Ming Cheng, Minghao Liu, Zhihong Zhang, and Xin Li. Lidar-based localization using universal encoding and memory-aware regression. *Pattern Recognition*, 128:108685, 2022.

[12] Huan Yin, Xuecheng Xu, Sha Lu, Xieyuanli Chen, Rong Xiong, Shaojie Shen, Cyrill Stachniss, and Yue Wang. A survey on global lidar localization: Challenges, advances and open problems. *International Journal of Computer Vision*, 132(8):3139–3171, 2024.

[13] Yuyang Yang, Wen Li, Sheng Ao, Qingshan Xu, Shangshu Yu, Yu Guo, Yin Zhou, Siqi Shen, and Cheng Wang. Raloc: Enhancing outdoor lidar localization via rotation awareness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3304–3313, 2025.

[14] Olivia Wiles, Sebastien Ehrhardt, and Andrew Zisserman. Co-attention for conditioned image matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15920–15929, June 2021.

[15] Fereidoon Zangeneh, Leonard Bruns, Amit Dekel, Alessandro Pieropan, and Patric Jensfelt. Conditional variational autoencoders for probabilistic pose regression. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2794–2800. IEEE, 2024.

[16] José Mena, Oriol Pujol, and Jordi Vitrià. A survey on uncertainty estimation in deep learning classification systems from a bayesian perspective. *ACM Computing Surveys (CSUR)*, 54(9):1–35, 2021.

[17] Wen Li, Yuyang Yang, Shangshu Yu, Guosheng Hu, Chenglu Wen, Ming Cheng, and Cheng Wang. Diffloc: Diffusion model for outdoor lidar localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15045–15054, 2024.

[18] Shengguang Wu, Fan-Yun Sun, Kaiyue Wen, and Nick Haber. Symmetrical visual contrastive optimization: Aligning vision-language models with minimal contrastive images. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30284–30297, 2025.

[19] Han Wang, Yuxiang Nie, Yongjie Ye, Yanjie Wang, Shuai Li, Haiyang Yu, Jinghui Lu, and Can Huang. Dynamic-vlm: Simple dynamic visual token compression for videollm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20812–20823, 2025.

[20] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[21] Zhaoyang Huang, Yan Xu, Jianping Shi, Xiaowei Zhou, Hujun Bao, and Guofeng Zhang. Prior guided dropout for robust visual localization in dynamic environments. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2791–2800, 2019.

[22] Fereidoon Zangeneh, Amit Dekel, Alessandro Pieropan, and Patric Jensfelt. Quantifying epistemic uncertainty in absolute pose regression. In *Scandinavian Conference on Image Analysis*, pages 180–195. Springer, 2025.

[23] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE international conference on Robotics and Automation (ICRA)*, pages 4762–4769. IEEE, 2016.

[24] Wei Wang, Bing Wang, Peijun Zhao, Changhao Chen, Ronald Clark, Bo Yang, Andrew Markham, and Niki Trigoni. Pointloc: Deep pose regressor for lidar point cloud localization. *IEEE Sensors Journal*, 22 (1):959–968, 2021.

[25] 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.

[26] Shuai Chen, Xinghui Li, Zirui Wang, and Victor A Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *European Conference on Computer Vision*, pages 1–17. Springer, 2022.

[27] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *Conference on Robot Learning*, pages 1347–1356. PMLR, 2022.

[28] Sijie Wang, Qiyu Kang, Rui She, Wee Peng Tay, Andreas Hartmannsgruber, and Diego Navarro Navarro. Robustloc: Robust camera pose regression in challenging driving environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6209–6216, 2023.

[29] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10393–10401, 2020.

[30] Shangshu Yu, Xiaotian Sun, Wen Li, Chenglu Wen, Yunuo Yang, Bailu Si, Guosheng Hu, and Cheng Wang. Nidaloc: Neurobiologically inspired deep lidar localization. *IEEE Transactions on Intelligent Transportation Systems*, 25(5):4278–4289, 2023.

[31] Sijie Wang, Qiyu Kang, Rui She, Wei Wang, Kai Zhao, Yang Song, and Wee Peng Tay. Hypliloc: Towards effective lidar pose regression with hyperbolic fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5176–5185, 2023.

[32] Shangshu Yu, Cheng Wang, Yitai Lin, Chenglu Wen, Ming Cheng, and Guosheng Hu. Stcloc: Deep lidar localization with spatio-temporal constraints. *IEEE Transactions on Intelligent Transportation Systems*, 24 (1):489–500, 2022.

[33] Qing Li, Shaoyang Chen, Cheng Wang, Xin Li, Chenglu Wen, Ming Cheng, and Jonathan Li. Lo-net: Deep real-time lidar odometry. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8473–8482, 2019.

[34] Guangming Wang, Xinrui Wu, Zhe Liu, and Hesheng Wang. Pwclo-net: Deep lidar odometry in 3d point clouds using hierarchical embedding mask optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15910–15919, 2021.

[35] Jiuming Liu, Guangming Wang, Zhe Liu, Chaokang Jiang, Marc Pollefeys, and Hesheng Wang. Regformer: An efficient projection-aware transformer network for large-scale point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8451–8460, 2023.

[36] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[37] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[38] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[39] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[40] Martin Arjovsky and Leon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017.

[41] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *International Conference on Learning Representations*, 2022.

[42] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19830–19843, 2023.

[43] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[44] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

[45] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[46] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

[47] Sangyun Lee, Beomsu Kim, and Jong Chul Ye. Minimizing trajectory curvature of ode-based generative models. In *International Conference on Machine Learning*, pages 18957–18973. PMLR, 2023.

[48] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations*, 2024.

[49] Zhiqiang Jiang, Kun Dai, Ke Wang, Tao Xie, Zhendong Fan, Ruifeng Li, Peng Kang, and Lijun Zhao. Centra-net: A centralized network for visual localization spanning multiple scenes. *IEEE Transactions on Multimedia*, 2025.

[50] Le Chen, Weirong Chen, Rui Wang, and Marc Pollefeys. Leveraging neural radiance fields for uncertainty-aware visual localization. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6298–6305. IEEE, 2024.

[51] Changkun Liu, Shuai Chen, Yukun Zhao, Huajian Huang, Victor Prisacariu, and Tristan Braud. Hr-apr: Apr-agnostic framework with uncertainty estimation and hierarchical refinement for camera relocalisation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8544–8550. IEEE, 2024.

[52] Zeyi Sun, Ye Fang, Tong Wu, Pan Zhang, Yuhang Zang, Shu Kong, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. Alpha-clip: A clip model focusing on wherever you want. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13019–13029, June 2024.

[53] Dong Jing, Xiaolong He, Yutian Luo, Nanyi Fei, Wei Wei, Huiwen Zhao, Zhiwu Lu, et al. Fineclip: Self-distilled region-based clip for better fine-grained understanding. *Advances in Neural Information Processing Systems*, 37:27896–27918, 2024.

[54] Vicente Vivanco Cepeda, Gaurav Kumar Nayak, and Mubarak Shah. Geoclip: Clip-inspired alignment between locations and images for effective worldwide geo-localization. *Advances in Neural Information Processing Systems*, 36:8690–8701, 2023.

[55] David G. Shatwell, Ishan Rajendrakumar Dave, Sirnam Swetha, and Mubarak Shah. Gt-loc: Unifying when and where in images through a joint embedding space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1–11, October 2025.

[56] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

[57] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[58] Carlo Alberto Barbano, Benoit Dufumier, Edouard Duchesnay, Marco Grangetto, and Pietro Gori. Contrastive learning for regression in multi-site brain age prediction. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pages 1–4. IEEE, 2023.

[59] Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. Rank-n-contrast: learning continuous representations for regression. *Advances in Neural Information Processing Systems*, 36:17882–17903, 2023.

[60] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33: 7537–7547, 2020.

[61] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian conference on computer vision*, pages 548–562. Springer, 2012.

[62] 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.

[63] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, pages 1–31, 2024.

[64] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European conference on computer vision*, pages 685–702. Springer, 2020.

[65] Yan Li, Qiuhong Xu, and Wanrong Cao. Weak convergence of the split-step backward euler method for stochastic delay integro-differential equations. *Mathematics and Computers in Simulation*, 227:226–240, 2025.

[66] Nicholas Carlevaris-Bianco, Arash K Ushani, and Ryan M Eustice. University of michigan north campus long-term vision and lidar dataset. *The International Journal of Robotics Research*, 35(9):1023–1035, 2016.

[67] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 6433–6438. IEEE, 2020.

[68] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3075–3084, 2019.

[69] Wen Li, Chen Liu, Shangshu Yu, Dunqiang Liu, Yin Zhou, Siqi Shen, Chenglu Wen, and Cheng Wang. Lightloc: Learning outdoor lidar localization at light speed. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6680–6689, 2025.

[70] Emanuele Ledda, Giorgio Fumera, and Fabio Roli. Dropout injection at test time for post hoc uncertainty quantification in neural networks. *Information Sciences*, 645:119356, 2023.

[71] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3523–3532, 2019.

[72] Wei Li, Lin Ma, Haoze Chang, Xiangyun He, and Longteng Huang. Efficient collaborative perception with integrated uncertainty estimation via evidence regression. *IEEE Transactions on Intelligent Transportation Systems*, 26(11):20976–20989, 2025. doi: 10.1109/TITS.2025.3587766.

[73] Yan Xia, Yusheng Xu, Shuang Li, Rui Wang, Juan Du, Daniel Cremers, and Uwe Stilla. Soe-net: A self-attention and orientation encoding network for point cloud based place recognition. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11348–11357, 2021.

A Training Setup

A.1 Dataset Details

We evaluate our method on two large-scale outdoor benchmark datasets: Oxford Radar RobotCar [67] and NCLT [66]. Both datasets are available online. For each dataset, we list the corresponding data split as shown in Table 5. The Oxford dataset was recorded using an autonomous Nissan LEAF platform navigating for about 10km central route over 32 separate sessions. Our study exclusively utilizes the point cloud data from the left sensor. Similarly, the NCLT dataset was collected at the University of Michigan's North Campus via a Segway robotic platform. It consists of 27 traversals, each spanning approximately 5.5km, using a single Velodyne HDL-32E LiDAR for spatial scanning.

Table 5: **Dataset sequence details.** We list sequences for Oxford Radar RobotCar and NCLT.

Oxford RobotCar Dataset					NCLT Dataset				
Sequence	Length	Tag	Train	Test	Sequence	Length	Tag	Train	Test
11-14-02-26	9.37km	sunny	✓		2012-01-22	6.1km	overcast	✓	
14-12-05-52	9.22km	overcast	✓		2012-02-02	6.2km	sunny	✓	
14-14-48-55	9.04km	overcast	✓		2012-02-18	6.2km	sunny	✓	
18-15-20-12	9.04km	overcast	✓		2012-05-11	6.0km	sunny	✓	
15-13-06-37	8.85km	overcast		✓	2012-02-12	5.8km	sunny		✓
17-13-26-39	9.02km	sunny		✓	2012-02-19	6.2km	overcast		✓
17-14-03-00	9.02km	sunny		✓	2012-03-31	6.0km	overcast		✓
18-14-14-42	9.04km	overcast		✓	2012-05-26	6.3km	sunny		✓

A.2 Training Details

In our implementation, voxel sizes are set to 0.5 for NCLT and 0.25 for Oxford. The input point cloud sequence is a tuple of size 3 with a spacing of 2 frames. VeLoc is optimized for 150 epochs using AdamW with a peak learning rate of 5×10^{-4} under a one-cycle cosine annealing strategy. The SPAR module adopts a two-stage training scheme over 70 epochs. We initialize the learning rate at 5×10^{-5} for pre-training and 5×10^{-6} for fine-tuning. Pre-training uses a 5-epoch linear warmup followed by a constant learning rate, while fine-tuning uses a 5-epoch linear warmup followed by cosine annealing decay. The foundation model DINOv2 with ViT-S/16 backbone is used for feature learning.

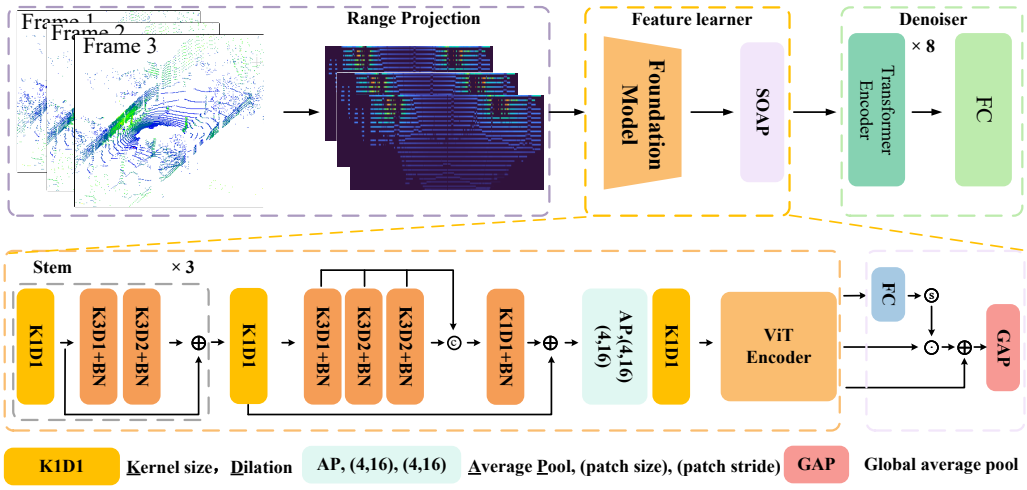


Figure 11: **Framework overview.** The layer elements K, D, and BN represent the kernel size, dilation rate, and batch normalization, respectively. Ⓢ in SOAP module indicates Sigmoid function.

The framework is shown in Figure 11. For the feature extractor backbone, we replace the patch embedding layer with convolutional stem in the standard ViT. The stem module used processes the

$5 \times H \times W$ input range image through a series of residual stages. Initially, three residual layers expand the feature dimension to 32, followed by subsequent layers that further increase the channel depth to 256 while maintaining the original spatial resolution. To satisfy the input constraints of the ViT, we apply average pooling to downsample the feature maps to a grid size of $[H/P_H] \times [W/P_W]$. A final 1×1 convolution layer then projects these features into a C_{stem}-dimensional space. In our implementation, with $[H, W] = [32, 512]$, $[P_H, P_W] = [4, 16]$, and $C_{stem} = 384$, the stem effectively produces 256 visual tokens. These tokens are then processed by a DINO-based ViT backbone to extract the deep feature map $F \in \mathbb{R}^{256\times384}$, excluding the classification token and the final output equals $F_P \in \mathbb{R}^{384}$. For the sampler, the Transformer consists of 8 encoder layers with 4 attention heads for feature aggregation. The latent embedding dimension is 512. The final fully connected layers of the sampler are configured as [512, 128, 6]. VeLoc was intentionally designed to mirror the DiffLoc baseline to ensure a fair comparison. Both methods share an identical feature pipeline and learned architecture. The difference lies in the generative formulation where VeLoc replaces diffusion-based refinement with conditional flow matching, together with the corresponding training objective and sampling procedure.

The geometric ADD metric[61], which quantifies the average point-wise displacement induced by the pose discrepancy, is calculated as:

$$\mathcal{L}_{\text{ADD}}(P_{\text{pred}}, P_{\text{gt}}; \mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}\in\mathcal{S}} \|(\mathbf{R}_{\text{gt}}\mathbf{x} + \mathbf{t}_{\text{gt}}) - (\mathbf{R}_{\text{pred}}\mathbf{x} + \mathbf{t}_{\text{pred}})\|_2, \tag{4}$$

where $P_{\text{pred}} = [\mathbf{R}_{\text{pred}}|\mathbf{t}_{\text{pred}}]$ and $P_{\text{gt}} = [\mathbf{R}_{\text{gt}}|\mathbf{t}_{\text{gt}}]$ denote the predicted and ground truth poses, respectively, and $\mathcal{S}$ represents the set of 3D points in the scan.

A.3 Representation details

In VeLoc, the 6-DoF pose is represented as $\mathbf{y} = [\mathbf{t}, \mathbf{r}] \in \mathbb{R}^6$. For the rotation component, we adopt an axis-angle representation $\mathbf{r}$, derived via the logarithmic map of the unit quaternion. This formulation maps the $SO(3)$ rotation manifold onto its corresponding tangent space $\mathfrak{so}(3)$. For a unit quaternion $\mathbf{q} = s + \mathbf{v}\hat{q}$, the logarithmic map is defined as:

$$\log(\mathbf{q}) = 0 + \hat{q}\frac{\mathbf{v}}{|\mathbf{v}|}\arccos(s) \tag{5}$$

where the resulting vector denotes the axis-angle representation. During the evaluation phase, the exponential map is employed to project the regressed vectors back into valid rotation matrices for rigorous metric computation.

Specifically, we enforce a quaternion hemisphere constraint by requiring the real part to be non-negative, which reduces ambiguity caused by quaternion double coverage. For the identity rotation, we directly set the logarithm to [0,0,0], thereby avoiding division-by-zero issues. Furthermore, we examine the target dataset for singular cases. Our task predicts the global pose, including rotation, rather than relative rotations. On the NCLT 2012-05-26 trajectory, we found no samples with rotation angles below 0.1° and only 21 samples below 1°, so near-singular cases are extremely rare. Therefore, although the axis-angle representation is theoretically singular, given our formulation, numerical preprocessing, and data distribution, we do not observe practical degradation in localization performance. Moreover, the subsequent optimization is carried out in Euclidean space, which helps alleviate singularity issues and numerical instability in practice.

This representation is advantageous as it bypasses the redundancy and unit-norm constraints inherent in quaternions, while providing a more continuous and stable space for regression compared to Euler angles, which are prone to gimbal lock. While $SO(3)$ manifold constraints are not explicitly enforced during the flow process, this serves as a computationally efficient and effective approximation for autonomous driving tasks. In such scenarios, the planar motion prior (stable altitude, roll, and pitch) ensures the numerical stability of the Euclidean approximation. This approach is consistent with all established baselines in the field. Nevertheless, we acknowledge that direct modeling on the $SO(3)$ Riemannian manifold constitutes a mathematically superior direction, and we aim to integrate such manifold-aware constraints in future extensions of this work.

B Additional Results

B.1 Localization Accuracy Comparison

We provide the detailed performance on both datasets here, including additional baselines beyond recent works as shown in Table 6 and Table 7. Consistent with the experimental results presented in the main text, our method achieves competitive performance on both datasets. More visualizations are shown in Section B.7.

Table 6: **Comparison with the state-of-the-art methods on NCLT dataset.** The mechanism (Mech.) S/A denotes the structure-based localization/absolute pose regression. We highlight the best and second-best results.

Methods	Mech.	2012-02-12	2012-02-19	2012-03-31	2012-05-26	Average [m/°]	Ranks
PNVLAD[7]	S	7.75/6.49	7.47/5.49	6.98/5.67	14.34/7.93	9.14/6.40	12/13
DCP[71]	S	9.84/6.84	8.27/5.16	8.94/5.96	15.62/7.99	10.67/6.49	14/14
SGLoc[9]	S	1.20/3.08	1.20/3.05	1.12/3.28	3.81/4.74	1.83/3.54	4/6
LightLoc[69]	S	0.98/2.76	0.89/2.51	0.86/2.67	3.10/3.26	1.46/2.80	3/3
PointLoc[24]	A	7.23/4.88	6.31/3.89	6.71/4.32	10.02/5.32	7.57/4.60	10/9
PosePN[11]	A	9.45/7.47	6.15/5.05	5.79/5.28	13.47/7.77	8.72/6.39	11/12
PosePN++[11]	A	4.97/3.75	3.68/2.65	4.35/3.38	9.59/4.49	5.65/3.57	8/7
PoseMinkLoc[11]	A	6.24/5.03	4.87/3.94	4.23/4.03	10.32/6.52	6.42/4.88	9/10
PoseSOE[11]	A	13.09/8.05	6.16/4.51	5.24/4.56	12.60/7.67	9.27/6.20	13/11
STCLoc[32]	A	4.91/4.34	3.25/3.10	3.75/4.04	8.67/5.23	5.15/4.18	7/8
NIDALoc[30]	A	4.48/3.59	3.14/2.52	3.67/3.46	6.60/4.56	4.47/3.53	6/5
HypLiLoc[31]	A	1.71/3.56	1.68/2.69	1.52/2.90	2.90/3.47	1.95/3.16	5/4
DiffLoc[17]	A	0.99/2.40	0.92/2.14	0.98/2.27	1.88/2.43	1.19/2.31	2/2
VeLoc(Ours)	A	0.82/2.44	0.73/2.12	0.77/2.23	1.78/2.39	1.02/2.29	1/1

Table 7: **Comparison with the baseline methods on the Oxford dataset.** The mechanism (Mech.) S/A denotes the structure-based localization/absolute pose regression. We highlight the best and second-best results.

Methods	Mech.	15-13-06-37	17-13-26-39	17-14-03-00	18-14-14-42	Average [m/°]	Ranks
PNVLAD[7]	S	18.14/3.28	24.57/3.08	19.93/3.13	15.59/2.63	19.56/3.03	14/13
DCP[71]	S	16.04/4.54	16.22/3.56	14.87/3.45	12.97/3.99	15.03/3.89	13/14
SGLoc[9]	S	3.01/1.91	4.07/2.07	3.37/1.89	2.12/1.66	3.14/1.88	2/8
LightLoc[69]	S	2.33/1.21	3.19/1.34	3.11/1.24	2.05/1.20	2.67/1.25	1/4
PointLoc[24]	A	12.42/2.26	13.14/2.50	12.91/1.92	11.31/1.98	12.45/2.17	11/10
PosePN[11]	A	14.32/3.06	16.97/2.49	13.48/2.60	9.14/1.78	13.48/2.48	12/12
PosePN++[11]	A	9.59/1.92	10.66/1.92	9.01/1.51	8.44/1.71	9.43/1.77	9/7
PoseMinkLoc[11]	A	11.20/2.62	14.24/2.42	12.35/2.46	10.06/2.15	11.96/2.41	10/11
PoseSOE[11]	A	7.59/1.94	10.39/2.08	9.21/2.12	7.27/1.87	8.62/2.00	8/9
STCLoc[32]	A	6.93/1.48	7.55/1.23	7.44/1.24	6.13/1.15	7.01/1.28	7/5
NIDALoc[30]	A	5.45/1.40	7.63/1.56	6.68/1.26	4.80/1.18	6.14/1.35	6/6
HypLiLoc[31]	A	6.88/1.09	6.79/1.29	5.82/0.97	3.45/0.84	5.74/1.05	5/3
DiffLoc[17]	A	3.57/0.88	3.65/0.68	4.03/0.70	2.86/0.60	3.53/0.72	4/2
VeLoc(Ours)	A	3.30/0.92	3.76/0.62	3.94/0.59	2.66/0.55	3.42/0.66	3/1

Statistical analysis. To provide a more comprehensive evaluation of the performance, we report the statistical results in the following section. Mean and standard deviation (std.) of translation and rotation errors across all test samples for each trajectory are reported in Table 8.

Statistical results indicate that VeLoc(Ours) achieves more consistent predictions, as evidenced by the lower or equivalent std. across most of the tested trajectories. This stability suggests that VeLoc provides more robust localization compared to the diffusion baseline. Even in specific instances where our method lags behind slightly, it maintains on-par robustness, demonstrating its reliability in complex environments.

Additionally, we conducted statistical tests to provide further empirical evidence supporting our findings, as detailed in Table 9. We employ the following metrics: Win Rate (%) represents the

Table 8: **Quantitative comparison of translation and rotation errors.** Mean and standard deviation of translation errors (m) and rotation errors (deg) are reported for DiffLoc and our VeLoc across eight test trajectories.

Method	15-13-06-37	17-13-26-39	17-14-03-00	18-14-14-42	2012-02-12	2012-02-19	2012-03-31	2012-05-26
DiffLoc(Trans)	3.57±3.38	3.65±1.79	4.01±2.31	2.86±1.97	0.99±1.13	0.92±0.90	0.98±3.32	1.88±10.20
VeLoc(Ours)(Trans)	3.30±2.78	3.76±1.61	3.95±1.64	2.67±1.30	0.81±1.06	0.74±1.07	0.77±3.56	1.78±10.80
DiffLoc(Rot)	0.89±3.12	0.69±0.98	0.70±0.51	0.60±1.23	2.40±2.55	2.14±2.09	2.27±2.57	2.43±3.85
VeLoc(Ours)(Rot)	0.92±3.02	0.63±1.16	0.59±0.41	0.56±0.43	2.43±2.62	2.12±2.15	2.23±2.29	2.39±3.52

percentage of total frames where VeLoc achieved a lower localization error than the baseline. Mean Reduc. (m) is the mean signed error reduction in error between the baseline and VeLoc. p-value (Paired t) is the result of a Paired t-test used to determine if the performance difference is statistically significant. A very small p-value (typically < 0.05) indicates that the improvement is not due to random chance. When computing the p-value, the localization results are sorted by timestamp to ensure that the errors calculated between relative estimates are derived from the same LiDAR scan. All data are measured with the open-source scipy library to avoid confusion.

Table 9: **Statistical significance analysis of translation error reduction.** Win rate, mean reduction of translation error (m), and paired t-test p-values are reported for VeLoc compared with DiffLoc on the NCLT and Oxford datasets.

Dataset	Sequence	Frames	Win Rate (%)	Mean Reduc. (m)	p-value (Paired t)
NCLT	2012-02-12	25,546	63.87	0.1678	1.9×10^{-139}
	2012-02-19	26,543	64.41	0.1917	< 0.001
	2012-03-31	26,205	66.30	0.2064	5.6×10^{-284}
	2012-05-26	26,441	60.47	0.1017	5.8×10^{-4}
	Overall	**104,735**	**63.76**	**0.1668**	**Significant**
Oxford	15-13-06-37	36,109	55.49	0.2714	< 0.001
	17-13-26-39	42,778	49.65	-0.1135	4.1×10^{-15}
	17-14-03-00	43,705	51.78	0.0580	1.4×10^{-9}
	18-14-14-42	37,109	56.41	0.1927	7.1×10^{-120}
	Overall	**159,701**	**53.13**	**0.0916**	**Significant**

Table 9 presents a comprehensive statistical analysis of VeLoc's performance relative to the SOTA APR baseline. On the NCLT dataset, our method achieves an overall win rate of 63.76% and a mean error reduction of 0.1668m. Similarly, on the Oxford dataset, VeLoc maintains competitive performance with a 53.13% win rate. Notably, the p-values across most sequences are substantially lower than 0.001, providing strong empirical evidence that the accuracy improvements of VeLoc are statistically significant and not due to random variation. These results underscore the robustness and efficacy of our proposed method across diverse and large-scale environments.

Likelihood estimation. To evaluate how well the predicted distribution aligns with the true locations, we report negative log likelihood (NLL) as shown in Table 10.

Table 10: **Comparison of NLL calculated under ground truth.** Lower NLL value indicates better distribution approximation.

Dataset	NLL ↓		Error[m/°] ↓	
	VeLoc(Ours)	DiffLoc	**VeLoc(Ours)**	DiffLoc
NCLT	**-8.70**	6.87	**1.02/2.29**	1.19/2.31
Oxford	**-8.56**	6.95	**3.42/0.66**	3.53/0.72

The NLL calculated at the GT pose quantifies the alignment between the predicted distribution and the GT pose distribution. VeLoc achieves significantly lower NLL values compared to DiffLoc on both datasets, which suggests that our method provides better distribution approximation.

Performance Analysis. We observe that our method falls back on Oxford although its performance is superior on NCLT dataset. We note that the performance gap on Oxford stems from lower ground

truth quality [9], which has a greater impact on our method than on SCR, since SCR methods may be less sensitive to such noise because they exploit local geometric constraints and robust correspondence filtering. Conversely, on NCLT dataset which provides higher GT precision, our method consistently outperforms all baselines, demonstrating superior accuracy. We also note here that the performance is evaluated without uncertainty estimation.

B.2 Localization Latency and Parameter Comparison

In the main text, we compare VeLoc and DiffLoc in terms of sampling-step efficiency. To further examine the practical performance gap, we compare the end-to-end time cost as shown in Table 11. Feature extraction includes the total time for range image and voxel feature extraction. Notably, range image features are reused between VeLoc and SPAR, which significantly reduces the total computational overhead. Iterative optimization is evaluated under 10 sampling steps. SPAR introduces only the latency of pose encoding and feature fusion, resulting in a lightweight and computationally efficient design. The total end-to-end latency remains under 20 ms/sample, which provides a safety margin for real-time deployment. Minor numerical differences across tables arise from separate evaluation runs conducted when supplementing the experiments. While latency may exhibit marginal fluctuations across trials due to variations in GPU utilization and system overhead, the general efficiency profile remains consistent. Latency is measured with a batch size of 28.

Table 11: **Per-sample latency decomposition of VeLoc.** Mean processing time of each component (in ms) and total end-to-end latency are reported for the NCLT and Oxford sequences.

Dataset	Sequence	Feature Extr. (scene)	Feature Extr. (geometry)	Iterative Refinement	SPAR	**Total E2E**
NCLT	2012-02-12	5.33	1.74	0.68	0.12	7.87
	2012-02-19	5.34	1.67	0.68	0.13	7.82
	2012-03-31	5.31	1.64	0.63	0.13	7.71
	2012-05-26	5.38	1.72	0.67	0.13	7.90
	Mean	**5.34**	**1.69**	**0.67**	**0.13**	**7.83**
Oxford	15-13-06-37	5.38	1.14	0.63	0.14	7.29
	17-13-26-39	5.31	1.11	0.64	0.13	7.19
	17-14-03-00	5.32	1.03	0.67	0.13	7.15
	18-14-14-42	5.33	1.12	0.63	0.13	7.21
	Mean	**5.34**	**1.10**	**0.64**	**0.13**	**7.21**

In addition, we provide a more comprehensive efficiency comparison, covering FLOPs, MACs, NFEs, performance metrics, and actual latency, as reported in Table 12. We selected different NFE settings to maintain similar performance for fair comparison. As shown in Table 12, VeLoc achieves substantially higher overall inference efficiency.

Table 12: **Comparison of computational complexity and pose accuracy.** FLOPs, MACs, number of function evaluations (NFEs), and mean translation (m) / rotation (°) errors are reported for VeLoc and DiffLoc.

Method	FLOPs ↓ (G)	MACs ↓ (G)	NFEs ↓	Performance ↑ (m/°)
DiffLoc	50.66	25.33	15	3.51 / 0.72
VeLoc (Ours)	**50.2**	**25.1**	**2**	**3.44 / 0.71**

B.3 Failure Detection for Safe Deployment

To illustrate the failure detection ability of our proposed SPAR module, we benchmark our method against variance baseline following prior works as shown in Figure 12.

Quantitative Analysis on Heavy-Tailed Distributions. As discussed above, SPAR is not primarily intended to improve the average localization accuracy. Instead, its role is to identify rare but catastrophic localization outliers that dominate operational risk. We therefore perform a stress-test analysis on explicitly selected heavy-tailed sequences from both the Oxford and NCLT datasets. The following sequences exhibit pronounced long-tail behavior and provide a more meaningful testbed for evaluating SPAR.

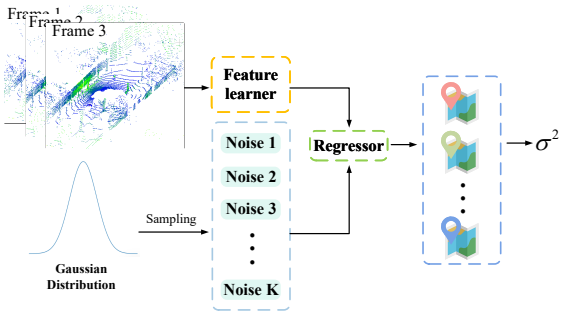


Figure 12: **Uncertainty estimation workflow.** We fix the model parameters and calculate the variance of the generated results conditioned on the same input scan from different noise samples.

Table 13: **Stress testing of SPAR on heavy-tailed sequences.** We report the 95th (P95) and 99th (P99) percentile translation errors to characterize the tail of the error distribution, together with extreme failure detection metrics.

Dataset	Heavy-tailed Sequence	P95 ATE (m)	P99 ATE (m)	$\text{AUROC}_{10\text{m}} \uparrow$	Recall @ 5% Budget ↑
Oxford	`15-13-06-37`	5.89	15.62	0.907	63.0%
NCLT	`2012-05-26`	3.53	20.10	0.911	49.4%

Metric definitions.

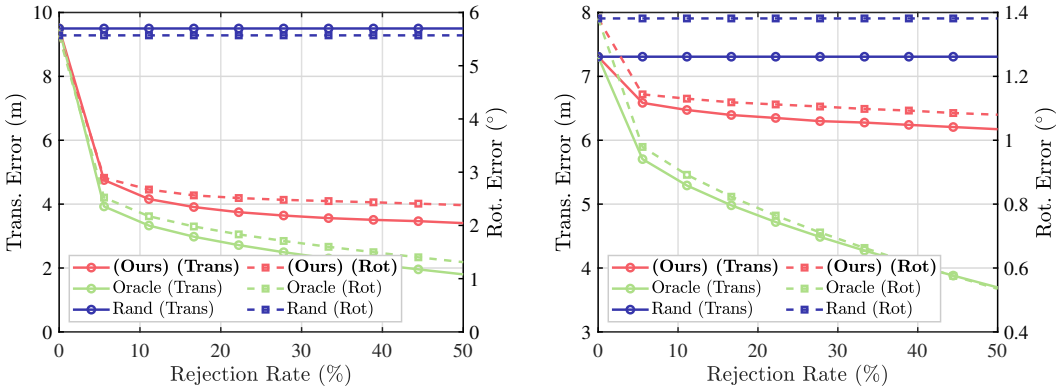
- **P95 and P99 translation errors.** These metrics evaluate the system's robustness in the long tail by characterizing its worst-case performance. Specifically, the P95 metric indicates the exact error threshold that 95% of the frames fall strictly *below*. In other words, it establishes an upper bound for the translation drift in the vast majority of operational conditions, revealing how severely the system degrades in the remaining 5% of extreme, challenging scenarios.
- **AUROC for failure detection.** While standard AUROC is designed to evaluate generic binary classifiers, $\text{AUROC}_{10\text{m}}$ specifically adapts this metric to assess failure detection in a continuous localization task by introducing a physical error threshold. To calculate this metric, we first binarize the continuous ground-truth localization performance. A frame is designated as the positive class, i.e., a failure if its actual translation error exceeds 10m, and as the negative class otherwise. Next, we utilize the system's estimated uncertainty as the continuous predicted risk score. By systematically sweeping a decision threshold across the full range of these risk scores, we compute the True Positive Rate (TPR) and False Positive Rate (FPR) at each operating point to construct the Receiver Operating Characteristic (ROC) curve. The Area Under this Curve ($\text{AUROC}_{10\text{m}}$) mathematically defines the probability that a randomly selected actual failure frame is assigned a higher predicted risk score than a randomly selected nominal frame.
- **Recall @ 5% Budget.** In practical autonomous driving and robotic deployment, continuous false alarms are impermissible. Simply alarming on all frames leads to accurate failure detection but is not allowed in real usage. Constantly triggering computationally expensive multi-sensor fusion or emergency braking would paralyze the system and severely degrade operational availability.

  Therefore, we impose an upper bound defined as a 5% intervention budget, which limits the system to evaluate only the TOP 5% of all frames. Under this restrictive formulation, the Recall @ 5% metric evaluates whether the proposed uncertainty estimation module is able to capture all severe trajectory drifts within this limited alarm quota.

As shown in Table 13, both datasets exhibit pronounced heavy-tailed behavior, with P99 errors reaching 15.62m on Oxford and 20.10m on NCLT. Under these stress-test conditions, SPAR remains effective at identifying catastrophic failures, achieving AUROC values above 0.90 on both sequences. Moreover, under a strict 5% alarm budget, SPAR successfully captures a substantial portion of the extreme outliers, recovering 63.0% of catastrophic failures on Oxford and 49.4% on NCLT.

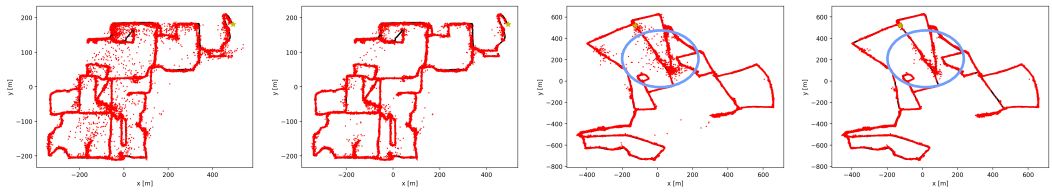
These results indicate that, while SPAR is not designed as a general-purpose detector for moderate localization errors, it provides strong practical utility in controlling tail risk.

Extended evaluation on NIDALoc. Similarly, beyond methods mentioned in the main paper, we extend our evaluation to NIDALoc[30] using the same two test trajectories. Given the deterministic nature of NIDALoc and the absence of dropout layers, comparison with the variance-based baseline is excluded. The results are shown in Figure 13 and Figure 14.



(a) NCLT (b) Oxford

Figure 13: **Error rejection curves on NIDALoc.** The solid lines and dashed lines denote translation (left Y-axis) and rotation (right Y-axis) errors, respectively. We benchmark our method against oracle and random (random rejection) on Oxford 15-13-06-37 trajectory and NCLT 2012-05-26 trajectory. Oracle denotes using actual error as confidence score, indicating theoretical lower bound.



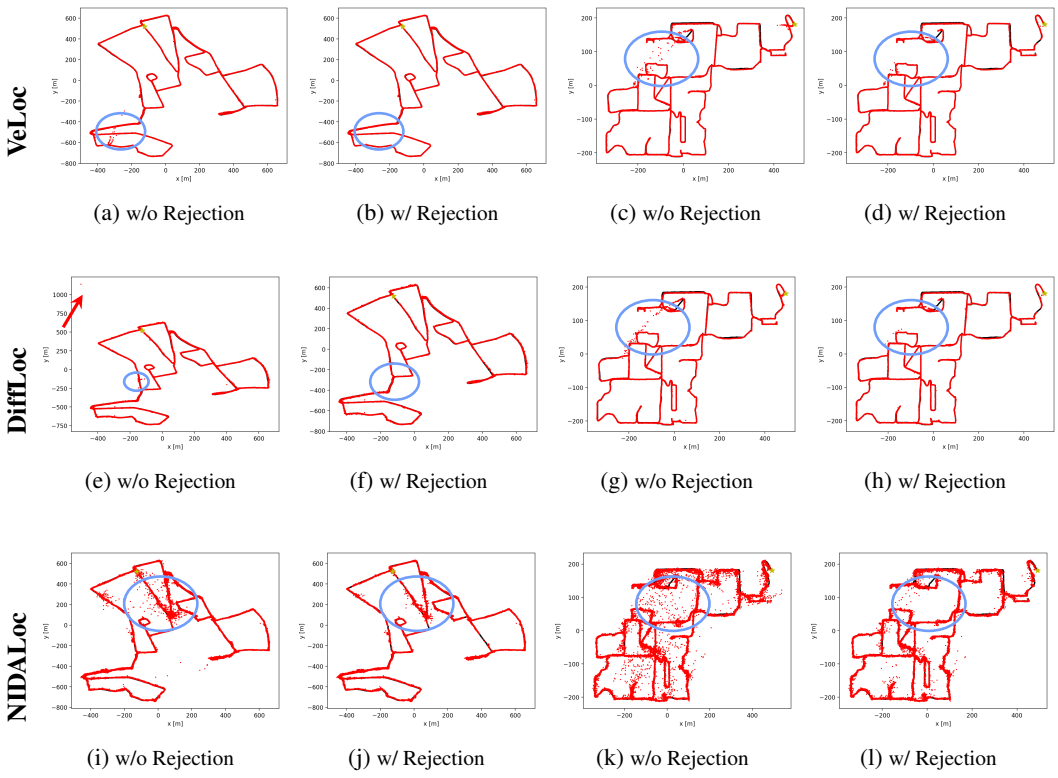
(a) NCLT w/o rejection (b) NCLT w/ rejection (c) Oxford w/o rejection (d) Oxford w/ rejection

Figure 14: **Trajectories visualization**. The trajectories of NIDALoc before and after the rejection on Oxford 15-13-06-37 trajectory and NCLT 2012-05-26 trajectory are shown.

It is evident that the SPAR module effectively identifies and filters out erroneous predictions, thereby enhancing overall localization accuracy. To better compare the performance of our SPAR method under various localization frameworks, we evaluated three distinct localization approaches. The corresponding visualizations are shown in the figure below.

Overall reliability comparison. Overall, based on further experiments conducted on VeLoc, DiffLoc, and NIDALoc, we validate the generalizability and localization-method agnostic nature of our proposed module. Based on these findings, we conclude that SPAR functions as a localization-method agnostic framework. It successfully generalizes across different localization methods by constructing an implicit memory of the spatial structure of the target scene via scan-pose embedding alignment. Our framework offers a versatile tool for enhancing system reliability. The explicit alignment capability enables direct, cross-method benchmarking, allowing SPAR to function as a decision-making supervisor in real-world applications, which can dynamically switch between different localization backbones to prioritize system safety.

Crucially, we emphasize that our method serves as a localization-method agnostic, plug-and-play module that requires no retraining or fine-tuning of the underlying localization backbone. Once trained on the target scene, it serves as an uncertainty estimator. This capability is particularly vital for deterministic architectures lacking stochastic layers (e.g., Dropout or BatchNorm), where



(a) w/o Rejection (b) w/ Rejection (c) w/o Rejection (d) w/ Rejection

(e) w/o Rejection (f) w/ Rejection (g) w/o Rejection (h) w/ Rejection

(i) w/o Rejection (j) w/ Rejection (k) w/o Rejection (l) w/ Rejection

Figure 15: **Trajectory visualization with SPAR module rejection across multiple sequences.** The first column indicates the sequence name. Each row compares the raw trajectory (w/o rejection) against the refined trajectory (w/ rejection) filtered by our post-hoc confidence estimation. Our method successfully detects the failure cases, especially in the circled areas with large errors.

conventional Monte Carlo sampling is inapplicable. Unlike alternative solutions that necessitate computationally expensive Deep Ensembles or complex Bayesian parameterization, our framework effectively mitigates these operational complexities.

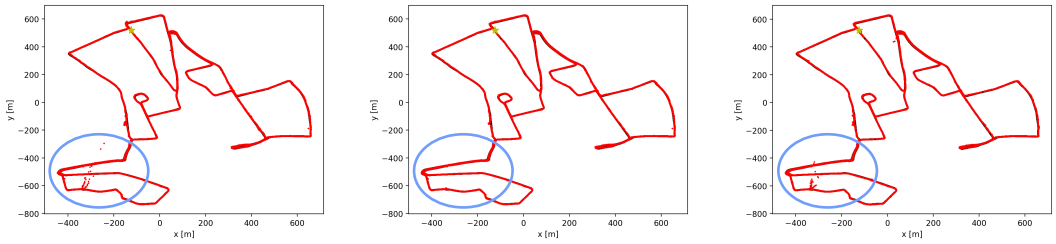
Comparison with other uncertainty estimation methods. Beyond the comparisons in the main text, we further analyze additional uncertainty estimation approaches. As previously discussed, conventional methods can be categorized as follows.

Heuristic methods [14, 49] typically attach task-specific uncertainty heads to a backbone network. These methods are generally not plug-and-play, as the heads take feature maps as input and are jointly optimized with the main task, making them tightly coupled with the localization architecture. Moreover, these heads are often designed under specific assumptions, such as particular pose representations and error metrics, which further limits their ability to generalize to localization models with different output formats. Consequently, adapting these methods to new models typically requires non-trivial redesign and retraining.

Distributional approaches [15, 22, 50, 72] model prediction errors using parametric distributions. Such modeling is often motivated by empirical observations such as histogram fitting, rather than rigorous theoretical guarantees [16]. While errors may appear approximately Gaussian on specific benchmarks, there is generally no formal guarantee that localization errors strictly follow a predefined distributional family under broader conditions. Some methods further adopt evidence regression or predefined parametric forms to improve performance, which may make them less transferable across different models.

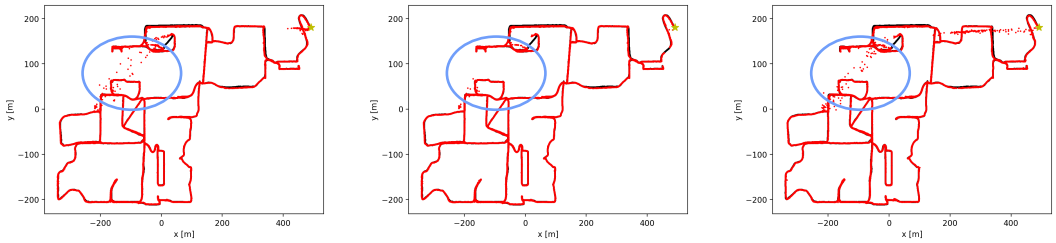
Meanwhile, variance-based techniques such as MC dropout and deep ensembles require multiple stochastic forward passes, making them computationally expensive for real-time applications. Furthermore, MC dropout is not directly applicable to deterministic architectures without dropout layers. To nevertheless provide a competitive baseline, we adopt a post-hoc dropout approximation, which has been shown to be an effective alternative [70]. By injecting a dropout layer at test time with all other sources of randomness fixed, the variability stems exclusively from the injected dropout. Under this protocol, the variance derived from surrogate dropout substantially underperforms our SPAR module. Moreover, we observe that both the injection position and the dropout ratio still require careful tuning to achieve optimal performance.

Importantly, SPAR enables uncertainty estimation in a single forward pass suitable for real-time deployment, without relying on model ensembles, thereby satisfying both efficiency and plug-and-play requirements. Further visualizations are shown below in Figure 16 and Figure 17.



(a) **Original VeLoc** (b) **Original VeLoc** + SPAR 10% (c) **Original VeLoc** + Dropout 10%

Figure 16: **Visualization on the Oxford dataset.** The starting frame is marked by a star. The black and red lines represent the ground truth and the predicted paths, respectively. The 10% samples with the lowest scores are rejected under different uncertainty estimation methods.



(a) **Original VeLoc** (b) **Original VeLoc** + SPAR 2% (c) **Original VeLoc** + Dropout 1%

Figure 17: **Visualization on the NCLT dataset.** The starting frame is marked by a star. The black and red lines represent the ground truth and the predicted paths, respectively. The 1% or 2% samples with the lowest scores are rejected under different uncertainty estimation methods since our method reaches SOTA on NCLT dataset.

Analysis of False Negatives in SPAR Training. Contrastive learning treats all non-matched samples as negatives. In place recognition, however, a robot may revisit previously observed locations within one traversal, so two scans that are geographically close to each other can be treated as a negative pair. Such false negatives are a well-known issue in contrastive learning and may harm the learned embedding space. We note here that SPAR adopts a strict one-positive formulation without any spatial filtering. Specifically, the GT pose is the only positive pose for each scan (query). This design is justified by three complementary aspects, i.e., the formulation choice at the pipeline level, a quantitative measurement of the false-negative ratio, and the practical engineering considerations.

First, the current frame is defined as the only positive sample for each query to maintain precise and unambiguous pose supervision. Treating nearby revisits as additional positives may introduce tolerance to pose discrepancies, which could weaken SPAR's intended discrimination of unreliable scan-pose alignments.

Second, to quantitatively assess the ratio of false-negative samples in the actual training pipeline, we calculate the false-negative rate by strictly adhering to the actual training logic. These statistics are computed by replaying the same batch-sampling and FIFO-update procedure used in training, and thus characterize the negative-pair distribution.

For a batch of B current samples and Q valid queue entries, we concatenate the B in-batch poses and the Q queued poses. For every query, its matched pose is the sole positive and all remaining $B + Q - 1$ candidates are treated as negatives. We use the translation components of each pose to validate revisits and we reverse the training-set normalization to recover world coordinates in meters. For every negative scan-pose pair, we compute its 3D Euclidean distance. A negative sample with distance below 5 m is recorded as a false negative. We also use different translation-based thresholds in this analysis, and the corresponding sensitivity results are reported in Table 15. We count the exact directed logits used by the loss, and thus in-batch $q_i \to p_j$ and $q_j \to p_i$ are counted twice. The resulting false-negative ratio is calculated as follows:

$$\frac{N_{\text{FalseNeg,batch}} + N_{\text{FalseNeg,queue}}}{B(B-1+Q)} \tag{6}$$

The resulting training-pair statistics are listed in Table 14:

Table 14: Training-pair statistics on Oxford and NCLT.

Dataset	Potential False-negative logits	Total negative logits	Ratio
Oxford	220,726	74,453,176	0.2965%
NCLT	143,369	50,872,276	0.2818%

Table 15: Sensitivity of the false-negative ratio with respect to the distance threshold.

Threshold	Oxford	NCLT
2 m	0.1504%	0.1286%
5 m	0.2965%	0.2818%
10 m	0.5725%	0.5534%

As shown in these tables, false-negative samples within the 5 m threshold account for fewer than 0.3% of the candidates actually used for contrastive training on both datasets. Given such a marginal ratio, the strict one-positive formulation introduces negligible label noise.

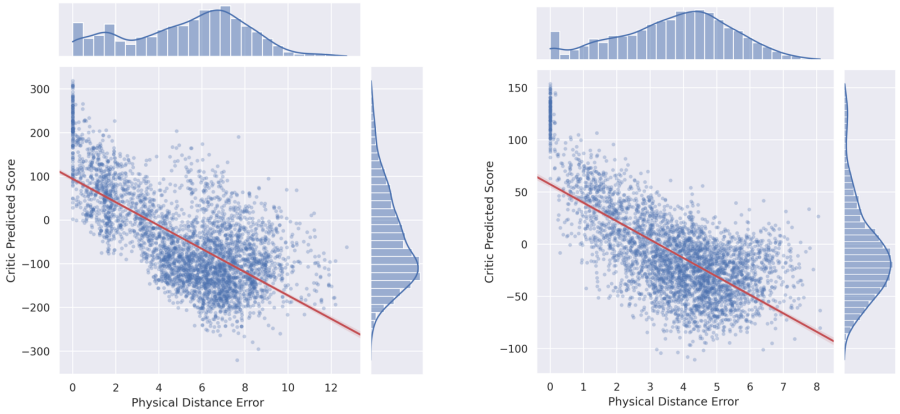
Third, the FIFO queue is a lightweight, standard, and dataset-agnostic mechanism. Applying distance-based filtering would require performing an additional geometry-aware check over the dynamically formed $B(B-1+Q)$ candidate logits at every training iteration, as well as threshold tuning for different datasets.

In summary, the strict one-positive formulation is a deliberate design choice that preserves unambiguous pose supervision and keeps the pipeline simple and dataset-agnostic while inducing negligible false-negative noise on both datasets. **Nevertheless**, explicit revisit-aware sampling remains an interesting direction, which we leave as a future extension.

B.4 Score-Error Alignment Verification

As discussed in the main paper, beyond merely rejecting unreliable candidates, a reliable uncertainty estimator should produce confidence scores that are negatively correlated with the pose error. To empirically validate the interpretability and reliability of the proposed scoring mechanism, we visualize the relationship between the predicted scores and the ground truth physical distance errors in Figure 18.

We calculate the pose error following [61] as in the main paper. We correlate the predicted scores with the average point distance error. In both cases, the joint distribution exhibits a negative correlation relationship, where higher confidence scores consistently correspond to lower geometric errors. The



(a) Oxford (b) NCLT

Figure 18: **Correlation visualization.** We visualize the correlation between physical distance error and predicted scores. (a) shows performance on the Oxford dataset, and (b) shows performance on the NCLT dataset. The red solid line indicates the linear regression fit. The marginal density plots reveal the distribution of pose errors and scores.

linear regression fit (indicated by the red solid line) confirms a negative correlation, suggesting that the learned metric effectively reflects the magnitude of pose error. This property is crucial for practical applications, as it allows for more informed decision-making based on the predicted confidence levels.

B.5 Multi-scale discrimination evaluation.

The SPAR module's ability to align pose and scan features inspires us to use it for map-database-free place recognition. The overall pipeline is shown in Figure 19.

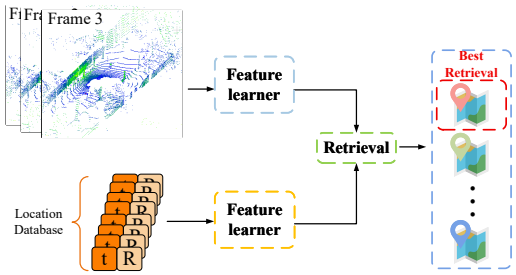


Figure 19: **The overall pipeline of using SPAR for place recognition.** The localization database is extremely lightweight and easy to maintain.

Additionally, it enables local discrimination for accuracy refinement beyond safe deployment assurance as shown in Table 16. Unlike prior methods[7, 73] which rely on explicit submaps, SPAR operates on a lightweight pose database. We construct the database using the poses from all training scans and adopt retrieval thresholds of 5m for NCLT and 25m for Oxford[54, 73]. For local discrimination, we generate 19 hard negatives by adding Gaussian perturbations with $\sigma_t = 0.05$ in normalized translation coordinates and $\sigma_r = 0.02$ in quaternion-log rotation coordinates. Unlike the coarse threshold-based metric used in place recognition, we adopt a stricter metric and directly recall the highest-scoring candidate, which requires distinguishing minimal geometric differences.

For place recognition, SPAR achieves robust R@1 scores of 91.11% and 94.74% for NCLT and Oxford datasets, respectively. While our results are slightly inferior to SOTA place recognition methods, we emphasize that our approach requires a lightweight pose database rather than burdensome map database. For fine-grained local discrimination, SPAR achieves a Top-1 Recall of 79.70% on NCLT and 68.64% on Oxford in this challenging task. The high Top-3 Recall shows that SPAR can reliably

Table 16: **Place recognition and local discrimination performance.** We report Recall@k scores of place recognition (Global) and fine-grained verification (Local) on both datasets.

Metric	Oxford		NCLT	
	Global	Local	Global	Local
R@1 (%) ↑	94.74	68.64	91.11	79.70
R@3 (%) ↑	95.47	89.75	97.01	96.89
R@5 (%) ↑	95.90	95.55	98.06	99.10

pick out the correct pose from similar candidates, validating the ability to distinguish minimal geometric differences.

Ablation Study We conduct ablation studies to comprehensively validate the effectiveness of our proposed training strategy as shown in Table 17. The overall results suggest that all components are essential for optimal performance, where the full configuration obtains the optimal performance balance across all metrics.

Table 17: **Ablation study of different loss components. Best** and second-best results are shown.

$\mathcal{L}_{\text{info}}^{\text{inter}}$	$\mathcal{L}_{\text{rank}}^{\text{inter}}$	$\mathcal{L}_{\text{info}}^{\text{intra}}$	$\mathcal{L}_{\text{rank}}^{\text{intra}}$	Oxford			NCLT		
				Global ↑	Local ↑	Pearson ↓	Global ↑	Local ↑	Pearson ↓
✓				75.63	61.17	-0.4480	81.39	70.67	-0.3159
✓	✓			84.17	64.24	-0.5893	85.04	74.47	-0.5347
✓	✓	✓		91.25	67.75	-0.5803	90.40	79.33	-0.4050
✓	✓	✓	✓	**94.74**	**68.64**	**-0.6890**	**91.11**	**79.70**	**-0.6856**

Based on the ablation results presented in Table 17, we validate the effectiveness of the proposed composite loss function on the NCLT and Oxford datasets. We establish a baseline by employing only the inter-scan contrastive loss $\mathcal{L}_{\text{infonce}}^{\text{inter}}$. While this configuration achieves coarse scan-pose alignment, it exhibits the lowest performance across all metrics. To explicitly supervise the score-error correlation, we incorporate the inter-scan ranking loss $\mathcal{L}_{\text{rank}}^{\text{inter}}$, resulting in significant improvements. Specifically, the Pearson correlation on NCLT improves markedly from -0.3159 to -0.5347, demonstrating that ranking constraints are crucial for aligning confidence scores with pose errors.

To further enhance fine-grained discrimination, we introduce $\mathcal{L}_{\text{infonce}}^{\text{intra}}$, leading to a substantial gain in retrieval accuracy. Notably, the global recall on the Oxford dataset increases by approximately 7% (reaching 91.25%), indicating that the effective utilization of hard negative mining enhances detail discrimination. Finally, the full model, which integrates the intra-scan ranking loss $\mathcal{L}_{\text{rank}}^{\text{intra}}$, achieves the best performance across all metrics. This configuration yields the most reliable uncertainty estimation, evidenced by the strongest negative Pearson coefficients and high recall rates on both tasks. In summary, these four components are complementary, collectively ensuring both high precision in localization and reliability in uncertainty quantification.

B.6 Discussion and Limitations.

We explicitly acknowledge that while SPAR is effective at capturing catastrophic outliers, it exhibits limited sensitivity to moderate localization drift (e.g., < 5m). This behavior stems from both the characteristics of the training data and a system-level design trade-off.

First, from a data perspective, the spatial resolution of the contrastive alignment is naturally limited. Given the scale of the training trajectories (around 10km per sequence) and the relatively large inter-frame distances typical of vehicle-mounted data collection, capturing fine-grained metric variations is challenging. Furthermore, the inherent ground-truth noise in large-scale datasets such as Oxford further limits the model's ability to reliably penalize micro-errors during training.

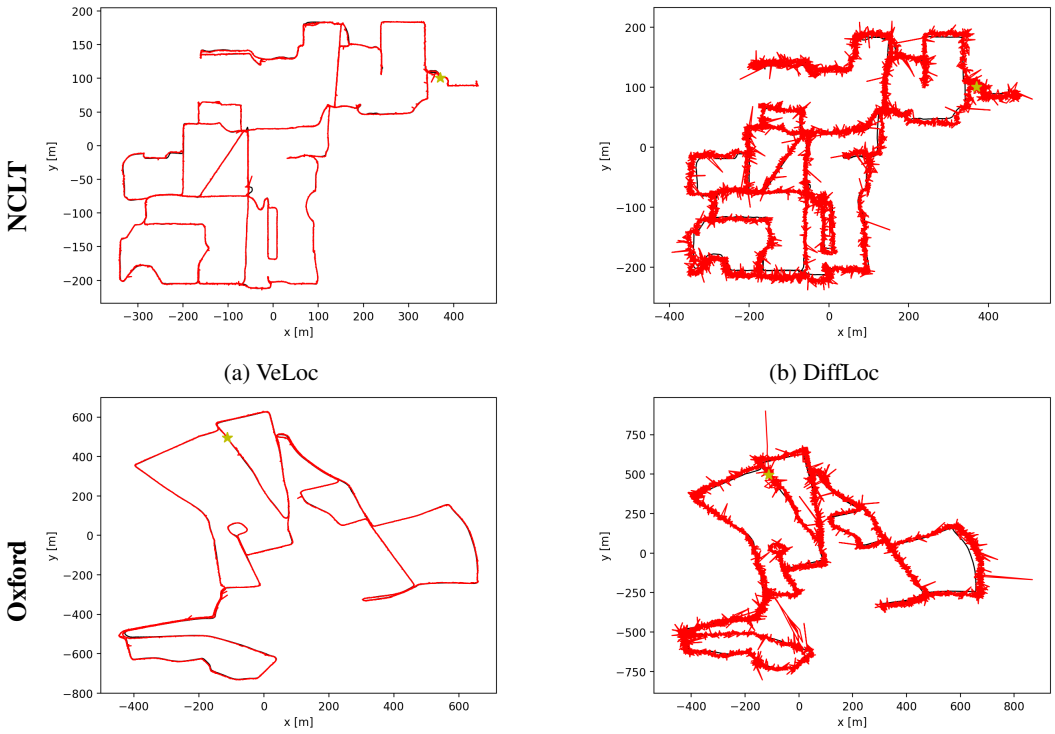
Second, this insensitivity also reflects an important design trade-off in the current formulation. SPAR is inherently more useful on sequences that contain heavy-tailed failures and macro-topological mismatches. Conversely, on highly stable and precise trajectories where extreme errors are rare, its discriminative advantage is naturally reduced, since there are fewer severe mismatches to detect.

In future work, we plan to address this limitation by exploring a hierarchical uncertainty framework. By coupling SPAR's macro-topological failure detection with traditional micro-level variance estimation, we aim to provide a more comprehensive safeguard that is sensitive to both local trajectory drift and catastrophic structural mismatches. We will also investigate more explicitly distance-aware hard-negative sampling to improve sensitivity to small drifts while preserving robustness to large failures.

In addition, we acknowledge that our localization method has not yet achieved the best performance on the Oxford dataset. In future work, we will further optimize the network design, refine the Oxford ground truth [9], and aim to achieve state-of-the-art performance. We also note here that our method employs a one-positive formulation, and explicit revisit-aware sampling remains an interesting direction, which we leave as a future extension.

B.7 Visualization

In the main text, to ensure a fair performance comparison, we uniformly adopted the optimal sampling step following DiffLoc baseline. However, our VeLoc method can localize within extremely low steps. Furthermore, the performance disparity between DiffLoc and VeLoc becomes even more pronounced when considering the jitter between adjacent frames.

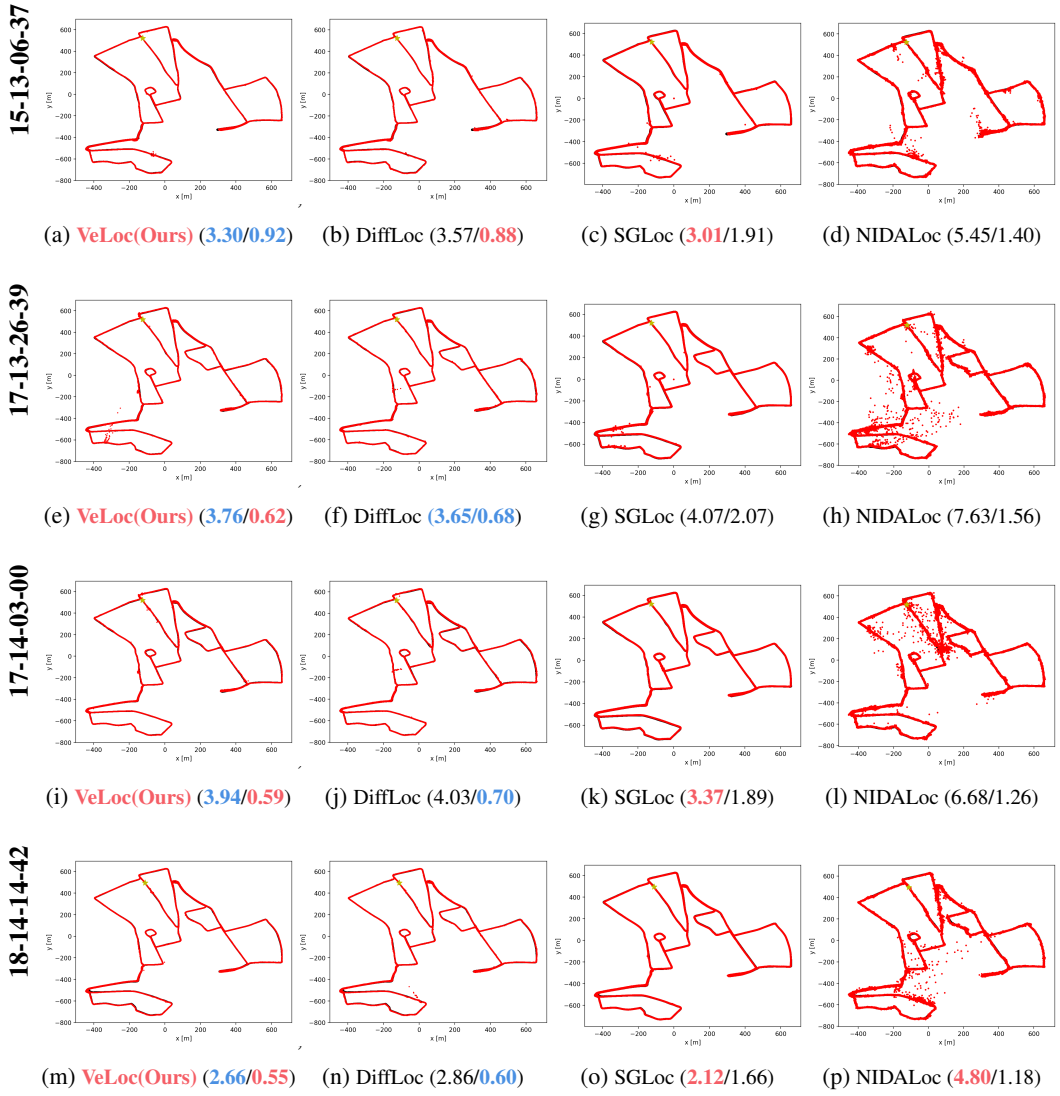


(a) VeLoc (b) DiffLoc

(c) VeLoc (d) DiffLoc

Figure 20: **Visualization on the NCLT and Oxford datasets.** The first row shows results on NCLT, and the second row shows results on Oxford. The starting frame is marked by a star. Both methods are benchmarked under 2 sampling steps. DiffLoc exhibits clear perturbations and drifting around the ground truth.

We show more visualization results of NIDALoc, DiffLoc, SGLoc, and the proposed VeLoc in Figure 21 and Figure 22 on the Oxford and NCLT datasets, respectively. It is important to note that on certain trajectories, VeLoc fails to achieve the best performance. However, our method achieves the optimal balance when considering the overall latency and accuracy.



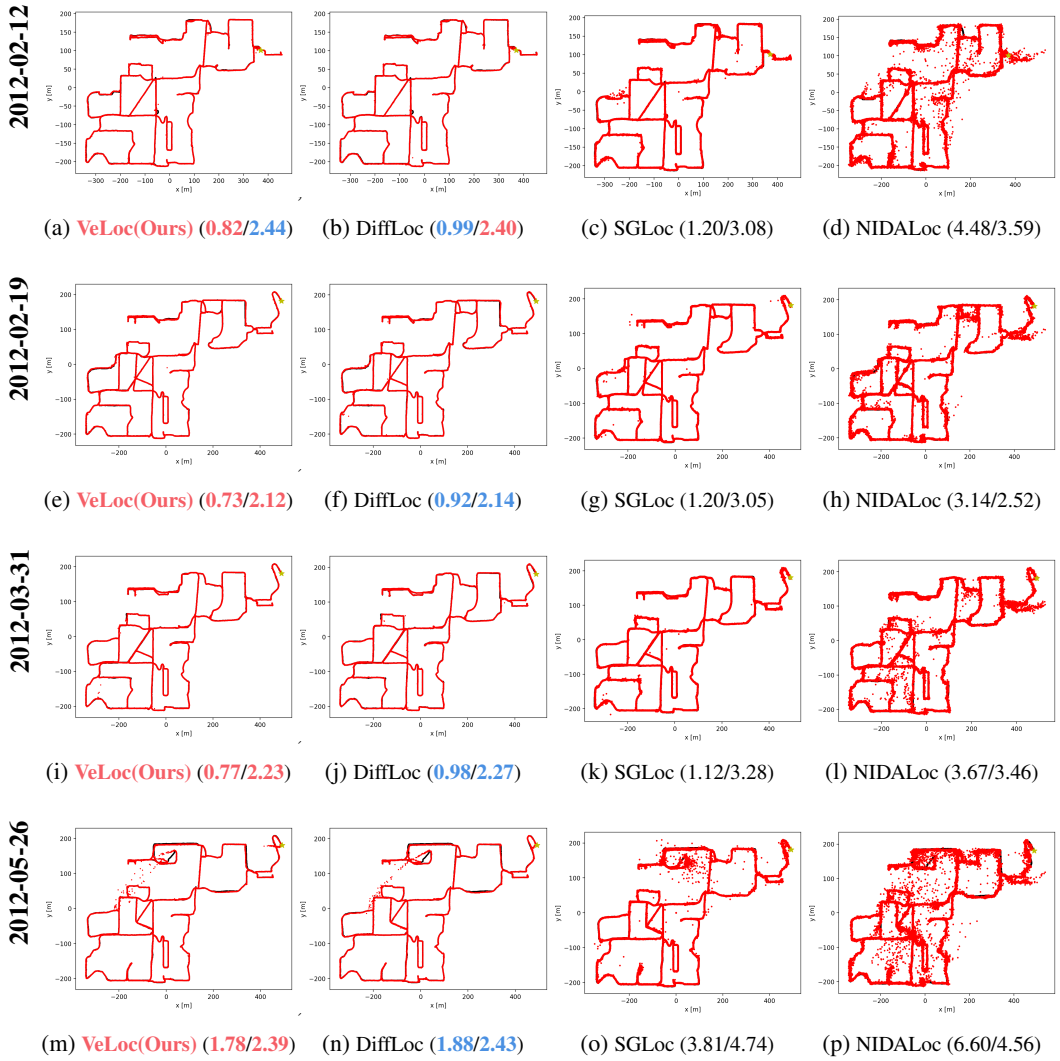
(a) VeLoc(Ours) (3.30/0.92) (b) DiffLoc (3.57/0.88) (c) SGLoc (3.01/1.91) (d) NIDALoc (5.45/1.40)

(e) VeLoc(Ours) (3.76/0.62) (f) DiffLoc (3.65/0.68) (g) SGLoc (4.07/2.07) (h) NIDALoc (7.63/1.56)

(i) VeLoc(Ours) (3.94/0.59) (j) DiffLoc (4.03/0.70) (k) SGLoc (3.37/1.89) (l) NIDALoc (6.68/1.26)

(m) VeLoc(Ours) (2.66/0.55) (n) DiffLoc (2.86/0.60) (o) SGLoc (2.12/1.66) (p) NIDALoc (4.80/1.18)

Figure 21: **Visualization results on 4 different Oxford sequences (rows).** The first column indicates the sequence name.



(a) VeLoc(Ours) (0.82/2.44) (b) DiffLoc (0.99/2.40) (c) SGLoc (1.20/3.08) (d) NIDALoc (4.48/3.59)

(e) VeLoc(Ours) (0.73/2.12) (f) DiffLoc (0.92/2.14) (g) SGLoc (1.20/3.05) (h) NIDALoc (3.14/2.52)

(i) VeLoc(Ours) (0.77/2.23) (j) DiffLoc (0.98/2.27) (k) SGLoc (1.12/3.28) (l) NIDALoc (3.67/3.46)

(m) VeLoc(Ours) (1.78/2.39) (n) DiffLoc (1.88/2.43) (o) SGLoc (3.81/4.74) (p) NIDALoc (6.60/4.56)

Figure 22: **Visualization results on 4 different NCLT sequences (rows).** The first column indicates the sequence name.

C Likelihood Estimation via Probability Flow

In this section, we provide the theoretical background for evaluating the negative log-likelihood (NLL) of the generated samples. We formulate both flow-matching and diffusion models within the unified framework of the probability flow ODE and the instantaneous change of variables theorem.

C.1 Probability Density Estimation in Continuous Normalizing Flows

The following derivation follows the continuity equation formulation in [45] and is included for completeness.

In sampling-based generative models, the intermediate state $\mathbf{x}_t$ at any time t represents a sample drawn from the marginal distribution $p_t(\mathbf{x})$. Since only empirical samples are accessible, the closed-form density $p_t(\mathbf{x})$ and its score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ cannot be computed directly via differentiation.

To obtain a differentiable trajectory for the time-evolving density, we adopt a fluid-dynamics perspective, treating $p_t(\mathbf{x})$ as a probability fluid flowing through space. Conservation of probability mass implies that the evolution of the density is governed by the continuity equation:

$$\frac{\partial p_t(\mathbf{x})}{\partial t} + \nabla_{\mathbf{x}} \cdot \mathbf{J}(\mathbf{x}, t) = 0, \tag{7}$$

where $\mathbf{J}(\mathbf{x}, t) = p_t(\mathbf{x})\mathbf{v}_t(\mathbf{x})$ is the probability current density, and $\mathbf{v}_t(\mathbf{x})$ is the time-dependent velocity field.

Consider a flow map $\Phi_t(\mathbf{x})$ governed by the ordinary differential equation (ODE) $\mathrm{d}\Phi_t(\mathbf{x})/\mathrm{d}t = \mathbf{v}_t(\Phi_t(\mathbf{x}))$, with the initial condition $\Phi_0(\mathbf{x}) = \mathbf{x}$. By applying the instantaneous change of variables formula alongside the continuity equation, we derive the rate of change of the log-density:

$$\frac{\mathrm{d}}{\mathrm{d}t} \log p_t(\Phi_t(\mathbf{x})) = -\operatorname{div} \mathbf{v}_t(\Phi_t(\mathbf{x})). \tag{8}$$

Integrating from time $t = 0$ (data distribution) to $t = 1$ (prior distribution) yields:

$$\log p_1(\Phi_1(\mathbf{x})) = \log p_0(\mathbf{x}) - \int_0^1 \operatorname{div} \mathbf{v}_t(\Phi_t(\mathbf{x}))\, \mathrm{d}t. \tag{9}$$

This allows us to evaluate the data likelihood $\log p_0(\mathbf{x})$ by reversing the integration. In practice, this is implemented by solving an augmented ODE:

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \Phi_t(\mathbf{x}) \\ \log p_t(\Phi_t(\mathbf{x})) \end{bmatrix} = \begin{bmatrix} \mathbf{v}_t(\Phi_t(\mathbf{x})) \\ -\operatorname{div} \mathbf{v}_t(\Phi_t(\mathbf{x})) \end{bmatrix}, \tag{10}$$

initialized with the test sample at $t = 0$. The divergence term $\operatorname{div} \mathbf{v}_t$ is estimated using Hutchinson's trace estimator for computational efficiency in high-dimensional spaces.

C.2 Probability Density Estimation in Diffusion Models

While diffusion models are typically formulated as stochastic differential equations (SDEs), exact likelihood computation requires a deterministic mapping. Here, we utilize the probability flow ODE formulation [43], ensuring consistency with the standard variance-preserving (VP) SDE schedule [37].

The forward process of a VP-SDE is given by:

$$\mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\, \mathrm{d}t + \sqrt{\beta(t)}\, \mathrm{d}\mathbf{w}. \tag{11}$$

It is proved [43] that there exists an equivalent deterministic probability flow ODE that shares the same marginal probability densities $\{p_t(\mathbf{x})\}_{t=0}^1$:

$$\mathrm{d}\mathbf{x} = \left[-\frac{1}{2}\beta(t)\mathbf{x} - \frac{1}{2}\beta(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right] \mathrm{d}t. \tag{12}$$

By substituting the score function approximation $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \approx -\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)/\sqrt{1-\bar{\alpha}(t)}$, we identify an induced velocity field $\tilde{\mathbf{v}}_t(\mathbf{x})$ for the diffusion model:

$$\tilde{\mathbf{v}}_t(\mathbf{x}) = -\frac{1}{2}\beta(t)\left(\mathbf{x} - \frac{\boldsymbol{\epsilon}_\theta(\mathbf{x}, t)}{\sqrt{1-\bar{\alpha}(t)}}\right). \tag{13}$$

With this induced velocity field, the NLL for the diffusion model can be computed using the same augmented ODE method described in Appendix C.1.

To ensure the continuous SDE aligns with the discrete-time DDPM trained with a linear schedule, we define $\beta(t)$ such that the marginal distributions match. The marginal distribution at time t is:

$$\mathbf{x}_t \mid \mathbf{x}_0 \sim \mathcal{N}\left(\sqrt{\bar{\alpha}(t)}\mathbf{x}_0, (1-\bar{\alpha}(t))\mathbf{I}\right), \tag{14}$$

where $\bar{\alpha}(t) = \exp\left(-\int_0^t \beta(s)\,\mathrm{d}s\right)$. In the discrete setting, β_t linearly increases from $\beta_{\min}$ to $\beta_{\max}$. Taking the limit of infinitesimal steps ($T \to \infty$), the corresponding continuous schedule becomes:

$$\beta(t) = \beta_{\min} + t(\beta_{\max} - \beta_{\min}), \quad t \in [0, 1]. \tag{15}$$

This schedule is used to compute the coefficients in the induced velocity field $\tilde{\mathbf{v}}_t(\mathbf{x})$.

C.3 Evaluation Metrics

Based on the derived log-probability, we evaluate the model using the negative log-likelihood (NLL). The negative log-likelihood (NLL) measures the alignment between the predicted distribution and the ground truth:

$$\mathrm{NLL} = -\frac{1}{N}\sum_{i=1}^{N} \log_2 p(\mathbf{p}_i \mid \mathbf{s}_i), \tag{16}$$

where $(\mathbf{p}_i, \mathbf{s}_i)$ denotes the i-th ground-truth pose-scan pair in the test set.

In the context of probability flow, the NLL measures the concentration of probability mass at the ground truth location. Minimizing NLL is equivalent to maximizing the likelihood that the generative trajectory converges exactly to the observed data.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: It accurately reflects our contributions and scope. The scope of the paper is illustrated in Section 1. We detail our method in Section 3. Experimental validation are detailed in Section 4

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: It has been discussed in the Appendix Section B.6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [No]

Justification: Our work incorporates only experimental results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Related information has been detailed in the paper. Section 3 includes brief information, whereas Section A.2 in the Appendix include more details. Moreover, code will be made publicly available when ready.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Codes will be released once upon ready. Instructions to faithfully reproduce the main experimental results are included in Section 3 and Section A.2 in the Appendix

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: They have been detailed in Section 4. Code will also be released once upon acceptance.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Appendix B.1 reports the mean and standard deviation of per-frame localization errors and paired statistical comparisons. The reported standard deviations describe variation across test frames, rather than across independent training runs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: This information has been detailed in Section B.2

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have read the code of ethics. Our research involves no human subjects or participants, and the usage of Oxford and NCLT datasets complies to their guidelines for privacy and ethical use cases. Information on reproducibility is provided in the Appendix. Our research currently involves no ethical considerations.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

Justification: Our work performs positive societal impact, such as its benefits for autonomous driving. We discuss it in Section 1. We have not found negative societal impact yet for LiDAR relocalization.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: Our paper poses no such risks. Our usage of data adhere to their official instructions and guidelines.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: They are credited and cited properly. Release information on datasets has been included in their paper. We properly respect the license and terms of use. We also leverage official implementation for some baselines. We have cited their paper and their paper contains release information.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: Our paper does not release new data assets. Codes will be released when ready. More detailed information such as exact commands for running the experiments, environment setup details and necessary data preprocessing steps will come with code release.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

Answer: [N/A]

Justification: Our paper does not involve crowdsourcing nor research with human subjects

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLM is used only for polishing *part* of the paper.

Guidelines:

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