GTR-Loc: Geospatial Text Regularization Assisted Outdoor LiDAR Localization

Shangshu Yu[1], **Wen Li**[2,3], **Xiaotian Sun**[2,3], **Zhimin Yuan**[4], **Xin Wang**[1], **Sijie Wang**[5], **Rui She**[6], **Cheng Wang**[2,3]*
[1]School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China
[2]Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, China
[3]Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, China
[4]School of Artificial Intelligence and Software Engineering, Nanyang Normal University, China
[5]Nanyang Technological University, Singapore
[6]Beihang University, China
yushangshu@cse.neu.edu.cn cwang@xmu.edu.cn

Abstract

Prevailing scene coordinate regression methods for LiDAR localization suffer from localization ambiguities, as distinct locations can exhibit similar geometric signatures — a challenge that current geometry-based regression approaches have yet to solve. Recent vision–language models show that textual descriptions can enrich scene understanding, supplying potential localization cues missing from point cloud geometries. In this paper, we propose GTR-Loc, a novel text-assisted LiDAR localization framework that effectively generates and integrates geospatial text regularization to enhance localization accuracy. We propose two novel designs: a Geospatial Text Generator that produces discrete pose-aware text descriptions, and a LiDAR-Anchored Text Embedding Refinement module that dynamically constructs view-specific embeddings conditioned on current LiDAR features. The geospatial text embeddings act as regularization to effectively reduce localization ambiguities. Furthermore, we introduce a Modality Reduction Distillation strategy to transfer textual knowledge. It enables high-performance LiDAR-only localization during inference, without requiring runtime text generation. Extensive experiments on challenging large-scale outdoor datasets, including QEOxford, Oxford Radar RobotCar, and NCLT, demonstrate the effectiveness of GTR-Loc. Our method significantly outperforms state-of-the-art approaches, notably achieving a 9.64%/8.04% improvement in position/orientation accuracy on QEOxford. Our code is available at `https://github.com/PSYZ1234/GTR-Loc`.

1 Introduction

Accurate and robust LiDAR localization, which estimates the position and orientation of LiDAR sensors, is fundamental to autonomous vehicles and robotics. Traditional approaches [17, 5, 8, 44, 11] typically perform localization by matching a query point cloud to a pre-built 3D map. Although effective, these methods often incur high storage costs for 3D maps [52] and substantial communication overhead [22], limiting their wide applications in large-scale outdoor environments.

End-to-end regression models have recently propelled LiDAR localization forward, overcoming limitations of previous methods by enabling deep networks to learn scene-specific representations. These models mainly fall into two categories based on different regression objectives: Absolute Pose

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

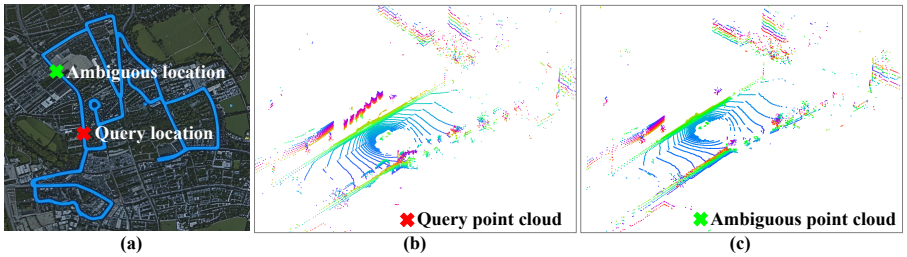

Figure 1: Illustration of LiDAR-localization ambiguities: panel (a) marks the true query location (red cross) and a spatially distinct ambiguous location (green cross), whereas panels (b) and (c) show the two sites' LiDAR point clouds, whose geometries appear nearly identical despite the distance.

Regression (APR) and Scene Coordinate Regression (SCR). APR models [41, 46, 48, 47, 40, 22, 12] directly regress the 6-DoF pose from point clouds, offering fast inference via compact architectures, yet often compromise accuracy due to limited geometric exploitation. Differently, SCR methods [23, 45, 21] predict world scene coordinates for each point, and then solve for the pose using RANSAC. SCR enforces geometric consistency during training, usually leading to improved performance.

However, current SCR methods suffer from a critical challenge—localization ambiguity arising from scene similarity, as shown in Fig. 1. This ambiguity occurs because distinct locations in large-scale outdoor environments often share highly similar local geometric structures. Consequently, methods (like SGLoc [23] and LightLoc [21]) relying only on local point cloud geometric features struggle to disambiguate visually similar, spatially distinct areas. This often results in erroneous coordinate predictions and subsequent localization failure. While LiSA [45] attempts to mitigate this ambiguity using semantic segmentation, geometrically similar regions often share similar semantic attributes, leaving the inherent problem fundamentally unsolved.

Recent advancements [1, 19, 13] in vision-language models highlight text's potent ability to enrich scene description and understanding. This insight motivates us to integrate textual cues that supply crucial scene localization information missing from common geometric data, thereby reducing localization ambiguities and improving performance. However, conventional text descriptions can be subjective, inconsistent across different times, or ambiguous for continuous observations [31, 50], making text-enhanced localization particularly challenging.

This paper proposes GTR-Loc, a novel text-assisted localization method that enhances SCR by generating and refining geospatial text regularization, thereby effectively mitigating ambiguities in LiDAR localization. Specifically, we propose two novel designs: Geospatial Text Generator (GTG) and LiDAR-Anchored Text Embedding Refinement (LATER). First, we propose a GTG to produce formatted text directly conveying discrete pose information. We partition geospatial positions and orientations into discrete districts and directions, then generate text based on the point cloud's current position and orientation. Unlike describing scene layout or the objects present, GTG directly constructs formatted text descriptions relevant to localization. Second, we propose LATER, a module that dynamically produces view-specific text embeddings. We learn a Transformer to leverage point cloud features to condition the instantiation of GTG, constructing refined LiDAR-text representations specific to the immediate view. By incorporating visual diversity within each district and direction category, it offers more effective regularization to reduce localization ambiguities. Finally, we introduce a Modality Reduction Distillation (MRD) strategy to enable LiDAR-only localization at inference, maintaining both effectiveness and efficiency. It distills textual regularization via a feature distillation module coupled with a distillation loss, transferring knowledge from LiDAR-text SCR to pure LiDAR SCR. Our contributions can be summarized as follows:

- We propose GTR-Loc, a novel text-assisted LiDAR localization framework. GTR-Loc is the first work to effectively design and integrate geospatial text descriptions as regularization to improve LiDAR SCR, leading to promising localization performance.
- We propose a Geospatial Text Generator and a LiDAR-Anchored Text Embedding Refinement module to dynamically create view-specific text descriptions focused on discrete pose information, providing significantly enhanced disambiguation capabilities for LiDAR localization.
- We devise a Modality Reduction Distillation strategy to enable LiDAR-only localization during inference. Extensive experiments on QEOxford [23], Oxford [2], and NCLT [26] datasets demon-

strate the great effectiveness of GTR-Loc, particularly outperforming state-of-the-art methods by 9.64%/8.04% on QEOxford.

2 Related Work

LiDAR Localization. Traditional LiDAR localization approaches typically aim to establish correspondences between the query point cloud and a pre-built 3D map. These approaches primarily achieve localization via map matching, such as PointNetVLAD [36], SOE-Net [43], SC^2-PCR++ [8], and TDM-RPMNet [49]. While potentially accurate, map-matching approaches suffer from expensive map storage, often at the terabyte to petabyte scale [52]. To address these limitations, regression-based localization is proposed, enabling inference without relying on pre-built 3D maps.

Absolute Pose Regression (APR). APR methods [15, 14, 32, 37, 33, 7, 34, 6] directly regress the 6-DoF pose from the input view (e.g., image or point cloud). Pioneering LiDAR APR works like PointLoc [41], employing PointNet++ [29] with an MLP head, demonstrate the approach's feasibility but show limitations in complex scenes. Subsequent works enhance APR with new architectures and loss functions. Notable examples include PosePN [46] with universal encoding and memory-aware regression; HypLiLoc [40] introducing cross-modal fusion and regression; and FlasMix [12] focusing on training acceleration. In the realm of multi-frame LiDAR localization, STCLoc [48] enforces spatio-temporal consistency across consecutive LiDAR scans, NIDALoc [47] draws inspiration from neurobiologically inspired mechanisms, and DiffLoc [22] refines poses through an iterative diffusion process. Nevertheless, APR's reliance on global scene representations can hinder the effective geometric encoding and potentially limit localization accuracy.

Scene Coordinate Regression (SCR). SCR methods [4, 39, 3, 38, 25] learns to predict per-point world coordinates, with RANSAC determining the final pose. SGLoc [23] first decouples localization into point correspondence regression and pose estimation. LiSA [45] distills semantic knowledge to enhance SCR. LightLoc [21] learns large-scale outdoor localization within 1 hour. Although SCR methods usually achieve higher localization accuracy than APR, they suffer from ambiguities arising from similar scenes, resulting in unreliable localization. To overcome the limitations of current methods, we propose GTR-Loc, a novel text-assisted LiDAR SCR framework. By incorporating and distilling geospatial textual regularization, we can effectively reduce localization ambiguities.

Vision-Language Models. Recent Large Language Models (LLMs), e.g., ChatGPT [27], LLaMA [35], and PaLM [9], have emerged as a promising way for understanding human languages. The success of LLMs has ignited interest in the Vision-Language (VL) research area. Foundational VL models like CLIP [31], COCOOP [51], and VL-Mamba [30] employ large-scale pre-training to align visual and textual representations.

These models also enable remarkable progress in diverse downstream tasks [1, 20, 19]. Since language and text can offer high-level scene descriptions, recent works explore their potential to address challenges in SLAM. LP-SLAM [50] and TextSLAM [18] first enable place recognition based on text labels within the SLAM system. Subsequently, Text2Pos [16], Text2Loc [42], and MNCL [24] propose using text to perform large-scale urban place recognition. Whereas existing approaches merely match text queries with images or point clouds, we embed structured text inside the pose estimation pipeline, enabling direct localization. We generate view-specific, pose-aware text descriptions as regularization for enhanced localization. Furthermore, we introduce a novel distillation strategy, enabling high-performance LiDAR-only localization during inference.

3 Method

3.1 Problem Formulation and Overview

Problem Formulation. The standard SCR framework for LiDAR-based localization aims to learn a mapping function, f_θ, parameterized by θ. Given an input LiDAR point cloud $P = \{p_i \in \mathbb{R}^3\}_{i=1}^N$, where each p_i is the point in the sensor's local frame, the objective is to predict the corresponding world scene coordinates $\hat{C} = \{\hat{c}_i \in \mathbb{R}^3\}_{i=1}^N$ for each point $\hat{c}_i$, such that $\hat{C} = f_\theta(P)$. Subsequently, the 6-DoF pose $[t, q]$ (a translation vector $t \in \mathbb{R}^3$ and a rotation vector $q \in \mathbb{R}^3$) can be estimated from the predicted point-to-point correspondences, typically using a RANSAC-based algorithm. In this paper, we formulate the SCR learning as multimodal regression, incorporating LiDAR point clouds

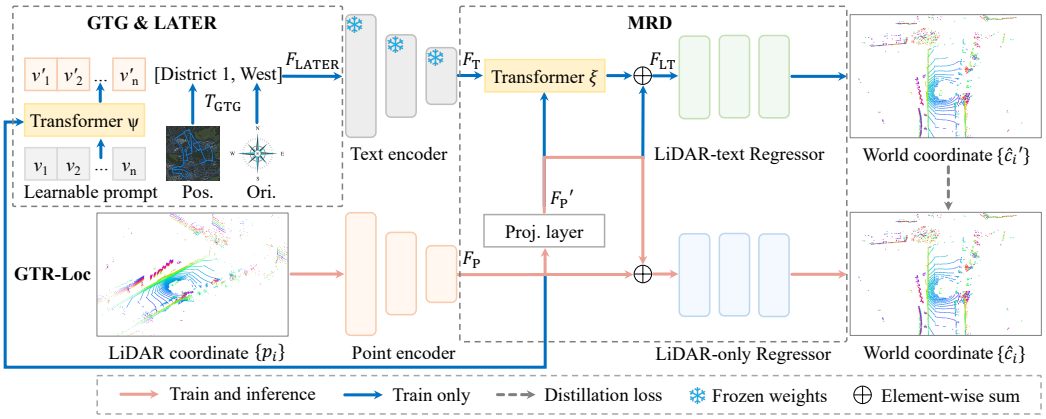

Figure 2: Overview of our method. GTR-Loc enhances LiDAR localization with text assistance to regularize SCR: a Geospatial Text Generator (**GTG**) provides discrete pose-aware text descriptions T_{GTG}, and a LiDAR-Anchored Text Embedding Refinement (**LATER**) module dynamically constructs view-specific text embeddings F_{LATER} conditioned on point features F_{P}. A Transformer ξ is employed to fuse multimodal features for LiDAR-text regression. Furthermore, a Modality Reduction Distillation (**MRD**) strategy enables LiDAR-only inference by distilling textural regularization.

with an additional input, geospatial text description T. Then, we learn a new mapping function g_ϕ, parameterized by ϕ, that uses both inputs to perform point regression, depicted as $\hat{C} = g_\phi(P, T)$.

Overview. This paper proposes GTR-Loc, a novel text-assisted LiDAR SCR framework, to address the challenge of ambiguities in LiDAR localization. Fig. 2 illustrates the network architecture of GTR-Loc. First, the point encoder extracts point cloud features F_{P} from the input LiDAR scan. Concurrently, we create discrete pose-aware text descriptions for each view to aid localization. Specifically, we propose two novel designs: Geospatial Text Generator (GTG) and LiDAR-Anchored Text Embedding Refinement (LATER) module. The proposed GTG (Sec. 3.2) generates formatted text descriptions T_{GTG} that directly convey discrete position and orientation information, which is relevant to localization. Then, the proposed LATER (Sec. 3.3) dynamically produces view-specific text embeddings F_{LATER} by incorporating current F_{P}, regularizing the SCR network for reducing ambiguities. Finally, a Transformer ξ combines text embeddings F_{T} with point cloud features $F_{\text{P}}^{'}$ and feeds the result F_{LT} to the LiDAR-text regressor. Furthermore, we propose Modality Reduction Distillation (Sec. 3.4) to distill geospatial text regularization, enabling LiDAR-only localization during inference. To supervise the model training, we adopt standard L1 losses (Sec. 3.5) to constrain the predicted and ground truth points. The SCR framework mostly follows the architecture of LightLoc [21], which consists of a multi-scale encoder and a regressor. We encode text using the text encoder of a pre-trained CLIP model [31]. Detailed descriptions are provided below.

3.2 Geospatial Text Generator

Traditional Vision-Language models [50, 18] leveraging text for SLAM often focus on generating descriptions of surrounding environments, e.g., scene layout or prominent objects. For example, a description of the scene point cloud is illustrated in Fig. 3 (b). While intuitive, relying on such descriptive text for LiDAR localization may face significant challenges. When revisiting the same location, dynamic changes in the scene can result in inconsistent textual descriptions. This inconsistency can significantly impede robust localization. In addition, describing scene content provides only indirect cues for localization, falling short of the direct position and orientation information required for 6-DoF pose estimation. To overcome these limitations, we propose a Geospatial Text Generator (GTG) to produce formatted pose-aware text descriptions that directly aid LiDAR localization.

Specifically, GTG generates concise descriptions grounded in predefined position and orientation partitions, as shown in Fig. 3 (c). We first divide the planar map into M distinct geographical districts to describe the LiDAR sensor's position. We define a mapping function $\mathcal{M} : x \to \{1, ..., M\}$, which maps a planar position $x \in \mathbb{R}^2$ of the input point cloud P to a discrete district identifier $z = \mathcal{M}(x)$. Concurrently, we discretize the compass directions into K directional bins to indicate the LiDAR

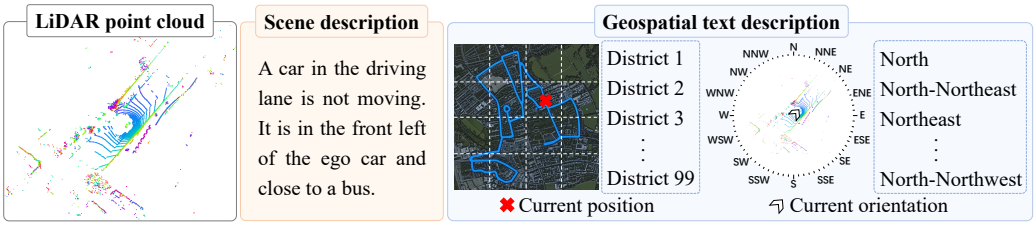

Figure 3: Scene description comparison. (a) Input LiDAR point cloud. (b) A free-form scene description, which is often subjective and verbose. (c) Our structured geospatial text description provides discrete pose cues for localization by encoding a district ID and a quantized direction.

sensor's heading. The direction set $\mathcal{D}$ includes cardinal (e.g., North), intercardinal (e.g., Northeast), and finer-grained bearings (e.g., North-Northeast) to capture more precise orientation. The orientation angle $\theta \in [0, 2\pi)$ of the input point cloud P is then mapped to a discrete direction bin d via a discretization function $\mathcal{O}$, such that $d = \mathcal{O}(\theta), d \in \mathcal{D}$. Finally, an example of the T_{GTG} template can be expressed as "`District 99, West-Southwest`."

These discretely determined identifiers, district z and direction d, represent the current geospatial state. Compared to conventional scene layout or object-centric descriptions, the proposed GTG provides more stable and informative cues for localization. For example, the constructed text, e.g., "District 99, West-Southwest", yields a signal that is stable and repeatable whether the scene changes. Conversely, free-form descriptions of the scene are subject to change due to moving objects and construction. Hence, the discrete pose-aware text in a standardized format offers clear advantages in aiding localization and solving ambiguities. The generated descriptions T_{GTG} are then input to the LATER module, which builds upon them to construct refined textual representations crucial for reducing scene ambiguities and thereby improving LiDAR localization performance.

3.3 LiDAR-Anchored Text Embedding Refinement

While GTG provides standardized, pose-aware text descriptions, its static prompts usually fail to capture the visual diversity within each district and direction category. For instance, consecutive LiDAR scans within the same district or direction may differ geometrically due to viewpoint shifts, occlusions, or environmental variations [16, 42, 24]. However, T_{GTG} yields identical geospatial text for these LiDAR point clouds. This consequently obscures fine-grained differences, leading to descriptive ambiguities that can impede high-accuracy localization. To address this, we propose a LiDAR-Anchored Text Embedding Refinement (LATER) module to refine T_{GTG} based on input point cloud features. Then, we can ensure that the refined text embeddings dynamically adapt to the unique characteristics of every input point cloud.

To be specific, we learn a Transformer decoder, parameterized by ψ, to generate conditional text embeddings for each LiDAR point cloud. As shown in Fig. 2, the Transformer takes two primary sets of inputs: n learnable prompts $\{v_1, v_2, ..., v_n\}$ and view-specific point cloud features F_{P}. The prompts are represented as learnable parameters using `nn.Parameter`. The feature F_{P} is directly extracted from the input point cloud. Within the Transformer, both self-attention and cross-attention mechanisms are utilized layer-wise to fuse these modalities. Self-attention stays inside v_n; cross-attention then connects v_n (as queries) to F_{P} (as keys/values). Finally, the refined text embedding F_{LATER} is constructed by combining the Transformer ψ's output $v_n^{'}$ with W_{GTG}, depicted as:

$$\{v_1^{'}, v_2^{'}, ..., v_n^{'}\} = Transformer_{\psi}(Q = \{v_1, v_2, ..., v_n\}, K, V = F_{\text{P}}), \tag{1}$$

$$F_{\text{LATER}} = \{v_1^{'}, v_2^{'}, ..., v_n^{'}, W_{\text{GTG}}\}, \tag{2}$$

where W_{GTG} is the word embedding of T_{GTG}.

The LiDAR anchoring mechanism refines the initial static prompt T_{GTG} with visual evidence from the input point cloud, effectively yielding view-specific textual embeddings F_{LATER}. It provides a more discriminative representation to distinguish subtle variations even within the same district or direction category. F_{LATER} not only aligns and adapts more closely to the input LiDAR scan, but

also delivers critical pose information for localization. Previous VLMs bridge vision and language via methods like COCOOP's learnable prompts [51], BLIP's contrastive-fusion learning [20], or Flamingo's gated cross-attention for injecting visual tokens into frozen LLMs [1]. However, they are designed for image understanding tasks such as image captioning or zero-shot classification, while LATER is specifically designed for LiDAR localization. Hence, it can effectively regularize SCR to reduce localization ambiguities. F_{LATER} is then processed by a frozen CLIP text encoder to produce the text embedding F_{T} for SCR.

3.4 Modality Reduction Distillation

Although the proposed geospatial text is effective, its construction relies on ground truth 6-DoF poses. This dependency is impractical for inference, as the pose is the very thing our localization task aims to estimate. Meanwhile, the absence of text or the presence of inaccurate text can lead to erroneous localization results. Hence, we introduce a Modality Reduction Distillation (MRD) strategy designed to transfer the knowledge from LiDAR-text SCR to pure LiDAR SCR. This allows us to achieve high-performance localization with only LiDAR inputs at inference, while retaining regularization imposed by the geospatial text.

As shown in Fig. 2, the multimodal regressor (LiDAR-text) acts as the teacher, while the pure point cloud regressor (LiDAR-only) acts as the student. We introduce a projection layer, a 3-layer MLP with identical hidden size, to transform point cloud features F_{P} into $F_{\text{P}}^{'}$. The text embedding F_{T} and $F_{\text{P}}^{'}$ are then integrated into a new Transformer decoder, parameterized by ξ, to be aggregated for the subsequent regression. The fused LiDAR-text features F_{LT} for SCR are depicted as:

$$F_{\text{LT}} = F_{\text{P}}^{'} + \alpha Transformer_{\xi}(Q = F_{\text{T}}, K, V = F_{\text{P}}^{'}). \tag{3}$$

Hence, the projection layer serves a dual role: it enables multimodal feature fusion for LiDAR-text SCR and, through a skip connection with F_{P}, contributes to LiDAR-only SCR as well. This allows for simultaneous feature enhancement and knowledge distillation. Finally, we define a distillation loss $\mathcal{L}_{\text{D}}$ to measure the discrepancy between the output of the teacher and student, represented as:

$$\mathcal{L}_{\text{D}} = \sum_{i=1}^{N} \|\hat{c}_i - \hat{c}_i^{'}\|_1, \tag{4}$$

where $\hat{c}_i$ and $\hat{c}_i^{'}$ denote the predicted world coordinates from the LiDAR-only regressor and LiDAR-text regressor, respectively. MRD enables the student regressor to mimic the teacher's refined outputs. The student thus inherits the text-assisted disambiguation, learning to narrow the search space using only its point cloud input. Once training is complete, the student SCR delivers accurate LiDAR-only localization at inference—no runtime text generation required.

3.5 Loss Functions

Our GTR-Loc architecture employs a multi-head learning framework that jointly optimizes LiDAR-text SCR and LiDAR-only SCR. The main optimization objective is to learn dense point predictions for localization. Hence, the proposed model is supervised by two loss functions during training:

$$\mathcal{L}_{\text{LT}} = \sum_{i=1}^{N} \|\hat{c}_i^{'} - c_i\|_1, \mathcal{L}_{\text{LO}} = \sum_{i=1}^{N} \|\hat{c}_i - c_i\|_1, \tag{5}$$

where c_i is the ground trurh world coordinates. The overall loss function $\mathcal{L}_{\text{SCR}}$ is a weighted summation of $\mathcal{L}_{\text{LT}}$, $\mathcal{L}_{\text{LO}}$, and the distillation loss $\mathcal{L}_{\text{D}}$ with a balancing weight β, represented as:

$$\mathcal{L}_{\text{SCR}} = \beta_1 \mathcal{L}_{\text{LT}} + \beta_2 \mathcal{L}_{\text{LO}} + \beta_3 \mathcal{L}_{\text{D}}. \tag{6}$$

4 Experiments

4.1 Experiment Settings

Datasets. We evaluate GTR-Loc on three commonly used large-scale outdoor datasets: Oxford Radar RobotCar (Oxford) [2], QEOxford [23], and NCLT [26]. **Oxford** is a large-scale urban dataset

Table 1: Quantitative results on the QEOxford dataset. We report the position error (m) and orientation error (°). Mechanism (Mech.) types are denoted as follows: MA for Multi-frame APR; SA for Single-frame APR; and SS for Single-frame SCR. Test Frames (TFs) denote the number of point cloud frames used during testing. We highlight the **best** and second-best results.

Methods	Mech.	TFs	15-13-06-37	17-13-26-39	17-14-03-00	18-14-14-42	Avg. [m/°]
STCLoc [48]	MA	3	5.14/1.27	6.12/1.21	5.32/1.08	4.76/1.19	5.34/1.19
NIDALoc [47]	MA	5	3.71/1.50	5.40/1.40	3.94/1.30	4.08/1.30	4.28/1.38
DiffLoc [22]	MA	3	**2.03/1.04**	**1.78/0.79**	**2.05/0.83**	**1.56/0.83**	**1.86/0.87**
PointLoc [41]	SA	1	10.75/2.36	11.07/2.21	11.53/1.92	9.82/2.07	10.79/2.14
PosePN [46]	SA	1	9.47/2.80	12.98/2.35	8.64/2.19	6.26/1.64	9.34/2.25
PosePN++ [46]	SA	1	4.54/1.83	6.44/1.78	4.89/1.55	4.64/1.61	5.13/1.69
PoseMinkLoc [46]	SA	1	6.77/1.84	8.84/1.84	8.08/1.69	6.56/2.06	7.56/1.86
PoseSOE [46]	SA	1	4.17/1.76	6.16/1.81	5.42/1.87	4.16/1.70	4.98/1.79
HypLiLoc [40]	SA	1	5.03/1.46	4.31/1.43	3.61/1.11	2.61/1.09	3.89/1.27
FlashMix [12]	SA	1	2.04/1.95	1.95/1.83	2.44/2.18	2.81/2.14	2.31/2.03
SGLoc [23]	SS	1	1.79/1.67	1.81/1.76	1.33/1.59	1.19/1.39	1.53/1.60
LiSA [45]	SS	1	0.94/1.10	1.17/1.21	0.84/1.15	0.85/1.11	0.95/1.14
LightLoc [21]	SS	1	0.82/1.12	0.85/1.07	0.81/1.11	0.82/1.16	0.83/1.12
GTR-Loc	SS	1	**0.77/1.02**	**0.77/1.01**	**0.67/1.01**	**0.80/1.07**	**0.75/1.03**

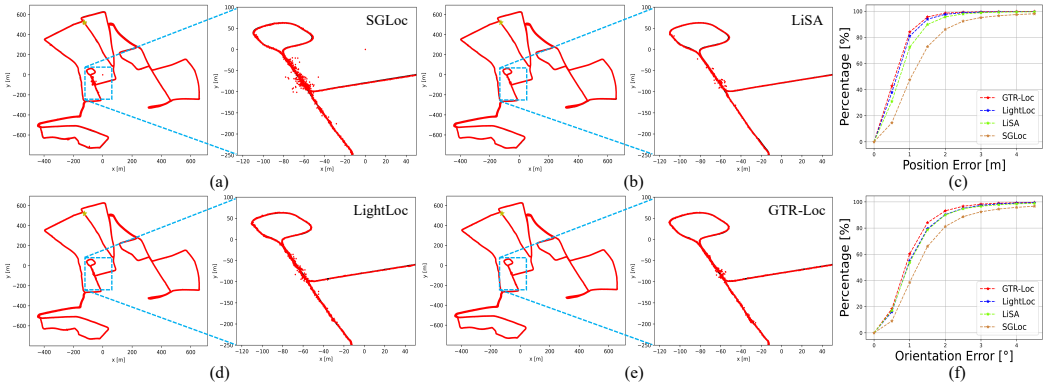

Figure 4: Visual comparisons on QEOxford. (a) (b) (d) (e): predicted trajectories (red) overlaid on ground truth (black); a star marks the starting position, and the blue box highlights a challenging road segment. (c) (f): cumulative error distribution curves for position (top) and orientation (bottom).

collected using a Nissan LEAF platform equipped with dual Velodyne HDL-32E sensors, providing dense LiDAR scans across multiple 10km trajectories within a 2km² city area. This dataset captures diverse variations in weather and traffic density. **QEOxford** is a quality-enhanced version of the Oxford dataset, more suitable for localization evaluation. The original Oxford dataset is further refined to mitigate the inherent errors in raw GPS/INS measurements. **NCLT** is a campus-scale dataset collected by a Segway robot with a Velodyne HDL-32E, covering 5.5km of trajectories within a 0.45km² area. This dataset encompasses both complex outdoor and indoor environments.

Baselines and Evaluation Metrics. To evaluate localization performance, we compare our approach against various LiDAR-based SCR and APR methods. For SCR, we include SGLoc [23], LiSA [45] and LightLoc [21], the latter being the state-of-the-art (SOTA) method. For APR, we evaluate against single-frame methods including PointLoc [41], PosePN [46], PosePN++[46], PoseSOE[46], PoseMinkLoc [46], HyLiLoc [40], and FlashMix [12], as well as multi-frame methods such as STCLoc [48], NIDALoc [47], and DiffLoc [22]. Following previous methods [22, 21], we report the position/orientation error [m/°] for each trajectory and the overall average across all trajectories.

Implementation Details. In this paper, we employ LightLoc [21] as the SCR backbone and a pre-trained CLIP [31] as the text encoder. Following LightLoc, we load the pre-trained encoder weights. We adopt the AdamW optimizer with a one-cycle learning rate schedule ranging from $5e-4$ to $5e-3$. The model is trained for 25 epochs on Oxford/QEOxford and 30 epochs on NCLT. Input

Table 2: Quantitative results on the Oxford dataset. The notations follow Tab. 1.

Methods	Mech.	TFs	15-13-06-37	17-13-26-39	17-14-03-00	18-14-14-42	Avg. [m/°]
STCLoc [48]	MA	3	6.93/1.48	7.55/1.23	7.44/1.24	6.13/1.15	7.01/1.28
NIDALoc [47]	MA	5	5.45/1.40	7.63/1.56	6.68/1.26	4.80/1.18	6.14/1.35
DiffLoc [22]	MA	3	**3.57/0.88**	**3.65/0.68**	**4.03/0.70**	**2.86/0.60**	**3.53/0.72**
PointLoc [41]	SA	1	12.42/2.26	13.14/2.50	12.91/1.92	11.31/1.98	12.45/2.17
PosePN [46]	SA	1	14.32/3.06	16.97/2.49	13.48/2.60	9.14/1.78	13.48/2.48
PosePN++ [46]	SA	1	9.59/1.92	10.66/1.92	9.01/1.51	8.44/1.71	9.43/1.77
PoseMinkLoc [46]	SA	1	11.20/2.62	14.24/2.42	12.35/2.46	10.06/2.15	11.96/2.41
PoseSOE [46]	SA	1	7.59/1.94	10.39/2.08	9.21/2.12	7.27/1.87	8.62/2.00
HypLiLoc [40]	SA	1	6.88/**1.09**	6.79/1.29	5.82/**0.97**	3.45/**0.84**	5.74/**1.05**
FlashMix [12]	SA	1	3.05/1.96	4.55/2.05	4.67/2.05	2.94/1.79	3.80/1.96
SGLoc [23]	SS	1	3.01/1.91	4.07/2.07	3.37/1.89	2.12/1.66	3.14/1.88
LiSA [45]	SS	1	2.36/1.29	3.47/1.43	3.19/1.34	1.95/1.23	2.74/1.32
LightLoc [21]	SS	1	2.33/1.21	3.19/1.34	3.11/1.24	2.05/1.20	2.67/1.25
GTR-Loc	SS	1	**2.29**/1.17	**3.07/1.21**	**2.99**/1.20	**2.00**/1.18	**2.59**/1.19

point clouds are voxel-downsampled with a voxel size of 0.25m on Oxford/QEOxford and 0.3m on NCLT. The number of districts z is set to 100, and the number of directions d is set to 16. α in Eq. 3 is set to 0.1. β_1 and β_2 in Eq. 6 are set to 1, β_3 is set to 0.1. GTR-Loc is implemented in PyTorch [28] and MinkowskiEngine [10]. All experiments are conducted on a single NVIDIA RTX 4090 GPU.

4.2 Comparison With State-of-the-Art Methods

Results on Oxford. We first evaluate GTR-Loc on the challenging QEOxford dataset. As shown in Tab. 1, GTR-Loc achieves SOTA accuracy among all single-frame SCR and APR methods, with a mean position/orientation error of 0.75m/1.03°. This performance significantly surpasses the previous leading SCR method, LightLoc (0.83m/1.12°), by 9.64%/8.04%, showcasing our effectiveness. Although our method processes only a single frame, it still surpasses DiffLoc, the SOTA multi-frame APR approach, by a substantial margin in positional accuracy. The results shown in Tab. 2 further demonstrate the superiority of our method (2.59m/1.19°) on the original Oxford dataset. This notable improvement highlights the effectiveness of integrating geospatial text regularization in resolving localization ambiguities, particularly in complex urban environments like Oxford.

To provide further insights into performance, we visualize representative predicted trajectories and cumulative error distribution curves on the QEOxford dataset. As illustrated in Fig. 4, our estimated trajectory on sequence 17-14-03-00 closely follows the ground truth trajectory over the entire sequence. As marked by blue boxes, GTR-Loc maintains stable and accurate tracking while competitor trajectories display systematic offsets, highlighting its robustness. The cumulative error distribution curves offer a quantitative summary supporting these observations. The sharper rise of our curves indicates consistently higher accuracy across most trajectory points. For example, over 85% of GTR-Loc's position errors fall below 1 m, compared to just 80% for LightLoc.

Results on NCLT. The NCLT dataset involves long-term data collection in a diverse setting, spanning both outdoor and indoor campus environments. As presented in Tab. 3, GTR-Loc achieves leading performance on position accuracy. Our approach yields mean position/orientation errors of 1.40m/2.62°, ranking first and second in position and orientation among single-frame SCR and APR methods. Our approach also delivers performance comparable to that of SOTA multi-frame APR methods, yet it operates using only a single frame during inference, thereby offering greater flexibility. This dataset presents unique difficulties like abrupt environmental transitions and varying structural complexity. GTR-Loc incorporates geospatial cues that remain informative across indoor and outdoor settings, resolving ambiguities in perceptually similar areas.

Runtime Analysis. We report the training (h)/inference (ms) time in Tab. 4. The entire model training process lasted approximately 4 hours, with a peak GPU memory consumption of around 10 GB. On the Oxford/QEOxford datasets, the average inference time per sample is 29ms (34 FPS), and on the NCLT dataset, it is 48ms (21 FPS). These speeds are well within the respective scanning frequencies of 20 Hz for Oxford/QEOxford and 10 Hz for NCLT, highlighting the model's ability to maintain high accuracy during real-time operation.

Table 3: Quantitative results on the NCLT dataset. The notations follow Tab. 1.

Methods	Mech.	TFs	2012-02-12	2012-02-19	2012-03-31	2012-05-26	Avg. [m/°]
STCLoc [48]	MA	3	4.91/4.34	3.25/3.10	3.75/4.04	7.53/4.95	4.86/4.11
NIDALoc [47]	MA	5	4.48/3.59	3.14/2.52	3.67/3.46	6.32/4.67	4.40/3.56
DiffLoc [22]	MA	3	**0.99/2.40**	**0.92/2.14**	**0.98/2.27**	**1.36/2.48**	**1.06/2.32**
PointLoc [41]	SA	1	7.23/4.88	6.31/3.89	6.71/4.32	9.55/5.21	7.45/4.58
PosePN [46]	SA	1	9.45/7.47	6.15/5.05	5.79/5.28	12.32/7.42	8.43/6.31
PosePN++ [46]	SA	1	4.97/3.75	3.68/2.65	4.35/3.38	8.42/4.30	5.36/3.52
PoseMinkLoc [46]	SA	1	6.24/5.03	4.87/3.94	4.23/4.03	9.32/6.11	6.17/4.78
PoseSOE [46]	SA	1	13.09/8.05	6.16/4.51	5.24/4.56	13.27/7.85	9.44/6.24
HypLiLoc [40]	SA	1	1.71/3.56	1.68/2.69	1.52/2.90	**2.29**/3.34	1.80/3.12
FlashMix [12]	SA	1	2.59/4.27	1.54/3.26	1.42/3.65	4.96/5.80	2.63/4.25
SGLoc [23]	SS	1	1.20/3.08	1.20/3.05	1.12/3.28	3.48/4.43	1.75/3.46
LiSA [45]	SS	1	0.97/**2.23**	0.91/**2.09**	0.87/**2.21**	3.11/**2.72**	1.47/**2.31**
LightLoc [21]	SS	1	0.98/2.76	0.89/2.51	0.86/2.67	3.10/3.26	1.46/2.80
GTR-Loc	SS	1	**0.95**/2.53	**0.82**/2.45	**0.82**/2.52	3.01/2.99	**1.40**/2.62

4.3 Ablation Study

Effects of Geospatial Text Generator. To comprehensively evaluate GTR-Loc, we first conduct ablation studies on the proposed Geospatial Text Generator (GTG). We report the average error for each dataset. As shown in Tab. 5, removing all proposed modules degrades performance from 0.75m/1.03° to 0.83m/1.12° (the vanilla model's result), highlighting the importance of geospatial text regularization. Then, we conduct experiments using different types of text descriptions. Specifically, we replace our geospatial text with a SOTA 3D scene description generator, TOD3Cap [13], that produces text describing scene layout and objects present. An example of the generated text for a scene is shown in Fig. 3 (b). Demonstrating no significant gains compared to the vanilla model on different datasets, this variant also remains markedly inferior to our full model. Human-like, free-form scene descriptions are inherently unstable for localization, as dynamic objects and environmental changes create inconsistent cues. In addition, they provide only indirect and noisy cues for pose estimation. Hence, the discrete pose-aware text generated by GTG delivers a more direct and effective localization cue than unconstrained scene descriptions, leading to better performance.

In addition, we conduct ablation studies by replacing the CLIP text encoder with embedding layers (ELs) to demonstrate the necessity of using textual representations. Specifically, we learn two separate, learnable embedding layers (`nn.Embedding`) initialized from scratch: one for the 100 district IDs and another for the 16 direction IDs. The resulting vectors are concatenated and passed through an MLP to match the feature dimension of the original CLIP embedding. The results shown in the table indicate that using simple learnable embeddings only leads to a small performance improvement. By processing geospatial text with CLIP, we can leverage pre-trained semantic priors, eliminating the need to learn complex semantic and spatial relationships from scratch.

Effects of District z and Direction d. To investigate the impact of spatial partitioning granularity in GTG, we conduct an ablation study by varying the number of districts z and directions d. We train and evaluate GTR-Loc with different configurations, varying z across values like 49, 100, 144, and d across values like 8, 16, 32. As shown in Tab. 6, localization accuracy improves as z increases from 49 to 100 and d from 8 to 16, indicating that finer geospatial discretization offers more discriminative contextual cues. However, further increasing z and d yields negligible gains, as excessive granularity fails to provide additional benefits in resolving localization ambiguities. The chosen values of z=100 and d=16 in this paper are appropriate for our method. This suggests that while sufficient granularity is needed for disambiguation, excessive partitioning may hinder performance.

Our method's partitioning scheme is both robust and scalable, making it practical for real-world, large-scale deployment. Its robustness stems from using an abstract coordinate grid rather than semantic environmental features, ensuring stability against long-term environmental changes like construction or seasonal variations. Results on the NCLT dataset, known for its data collection spanning several months, also demonstrate this. Furthermore, this design is also inherently scalable. The 10x10 uniform grid is illustrative and can be seamlessly extended to city-scale applications using

Table 4: Training (h)/Inference (ms) time.

Method	QEOxford	Oxford	NCLT
SGLoc	50/38	50/38	42/75
LiSA	53/38	53/38	44/75
LightLoc	1/29	1/29	1/48
GTR-Loc	4/29	4/29	3/48

Table 5: Ablation of GTG.

Method	QEOxford	Oxford	NCLT
vanilla	0.83/1.12	2.67/1.25	1.46/2.80
TOD3Cap	0.84/1.15	2.66/1.24	1.47/2.78
ELs	0.80/1.10	2.64/1.23	1.44/2.75
GTG	0.75/1.03	2.59/1.19	1.40/2.62

Table 6: Ablation of z and d.

Method	QEOxford	Oxford	NCLT
z=49, d=8	0.79/1.09	2.62/1.21	1.43/2.67
z=**100**, d=**16**	0.75/1.03	2.59/1.19	1.40/2.62
z=144, d=32	0.76/1.02	2.60/1.22	1.41/2.61

Table 7: Ablation of LATER.

Method	QEOxford	Oxford	NCLT
w/o LATER	0.78/1.10	2.62/1.22	1.43/2.70
Sum	0.76/1.06	2.62/1.21	1.41/2.66
Concatenate	0.77/1.09	2.61/1.22	1.41/2.68
Transformer	0.75/1.03	2.59/1.19	1.40/2.62

Table 8: Ablation of MRD.

Method	QEOxford	Oxford	NCLT
MRD (w/o text)	0.75/1.03	2.59/1.19	1.40/2.62
MRD (w text)	0.74/0.93	2.58/1.07	1.12/2.40

standard hierarchical grids (e.g., UTM). Our ablation study confirms its robustness as the number of partitions changes (e.g., from z=100 to z=49 or z=144).

Effects of LiDAR-Anchored Text Embedding Refinement. To evaluate the contribution of dynamically refining text embeddings using visual context, we conduct an ablation study on the proposed LiDAR-Anchored Text Embedding Refinement (LATER) module. We compare four distinct configurations: (1) a baseline using only the static prompt generated from GTG without LATER; (2) a simplified variant that replaces Transformer fusion with element-wise summation (3) another variant with feature concatenation fusion; and (4) our proposed LATER, which utilizes a Transformer to fuse text embeddings with point cloud features. The results reported in Tab. 7 clearly favor our proposed approach. Using only static prompts without LATER led to worse results, e.g., 0.78m/1.10° on QEOxford. The summation and feature concatenation fusion variant performs intermediately. The Transformer-based LATER achieves the best performance. It allows creating highly discriminative, scene-specific text embeddings using visual context for accurate localization.

Effects of Modality Reduction Distillation. We further evaluate the Modality Reduction Distillation (MRD) strategy in Tab. 8. We present ablation studies with text inputs, generated by ground truth poses, at inference. Results indicate that using text during inference usually leads to better performance, e.g., 0.74m/0.93° on QEOxford. However, generating text at inference can be impractical and introduce unwanted dependencies, since it depends on discrete pose estimates. The distilled model without text achieves performance nearly matching that of the text-assisted localization approach, e.g., 0.75m/1.03° on QEOxford. This demonstrates that MRD distills geospatial text regularization from the teacher effectively, enabling high-performance LiDAR-only localization during inference. In the Appendix, we provide additional ablation studies and visualizations for the Oxford and NCLT datasets, along with a discussion of limitations and future work.

5 Conclusion

This paper introduces GTR-Loc, the first text-assisted LiDAR localization framework that integrates geospatial text regularization into an SCR network to reduce localization ambiguities. The proposed geospatial text regularization consists of two components: a Geospatial Text Generator, which produces formatted, discrete pose-aware text descriptions, and a LiDAR-Anchored Text Embedding Refinement module, which dynamically constructs view-specific embeddings conditioned on current point cloud features. Furthermore, we introduce a Modality Reduction Distillation strategy to distill textual regularization knowledge, enabling high-performance LiDAR-only localization at inference time. Comprehensive experiments on challenging large-scale outdoor datasets, QEOxford, Oxford Radar Robotcar, and NCLT, demonstrate the effectiveness of GTR-Loc.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities under Grant N25XQD053. We would like to thank the anonymous reviewers for their valuable suggestions.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, volume 35, pages 23716–23736, 2022.

[2] 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 *ICRA*, pages 6433–6438, 2020.

[3] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *CVPR*, pages 5044–5053, 2023.

[4] Eric Brachmann and Carsten Rother. Visual camera re-localization from rgb and rgb-d images using dsac. *IEEE PAMI*, 44(9):5847–5865, 2021.

[5] Daniele Cattaneo, Matteo Vaghi, and Abhinav Valada. Lcdnet: Deep loop closure detection and point cloud registration for lidar slam. *IEEE TRO*, 38(4):2074–2093, 2022.

[6] Shuai Chen, Yash Bhalgat, Xinghui Li, Jia-Wang Bian, Kejie Li, Zirui Wang, and Victor Adrian Prisacariu. Neural refinement for absolute pose regression with feature synthesis. In *CVPR*, pages 20987–20996, 2024.

[7] Shuai Chen, Xinghui Li, Zirui Wang, and Victor Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *ECCV*, 2022.

[8] Zhi Chen, Kun Sun, Fan Yang, Lin Guo, and Wenbing Tao. Sc^2-pcr++: Rethinking the generation and selection for efficient and robust point cloud registration. *IEEE PAMI*, 45(10):12358–12376, 2023.

[9] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *JMLR*, 24(240):1–113, 2023.

[10] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, pages 3075–3084, 2019.

[11] Keyu Du, Hao Xu, Haipeng Li, Hong Qu, Chi-Wing Fu, and Shuaicheng Liu. Hybridreg: Robust 3d point cloud registration with hybrid motions. In *AAAI*, volume 39, pages 2789–2797, 2025.

[12] Raktim Gautam Goswami, Naman Patel, Prashanth Krishnamurthy, and Farshad Khorrami. Flashmix: Fast map-free lidar localization via feature mixing and contrastive-constrained accelerated training. In *WACV*, pages 2011–2020, 2025.

[13] Bu Jin, Yupeng Zheng, Pengfei Li, Weize Li, Yuhang Zheng, Sujie Hu, Xinyu Liu, Jinwei Zhu, Zhijie Yan, Haiyang Sun, et al. Tod3cap: Towards 3d dense captioning in outdoor scenes. In *ECCV*, pages 367–384, 2024.

[14] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, pages 5974–5983, 2017.

[15] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, page 2938–2946, 2015.

[16] Manuel Kolmet, Qunjie Zhou, Aljoša Ošep, and Laura Leal-Taixé. Text2pos: Text-to-point-cloud cross-modal localization. In *CVPR*, pages 6687–6696, 2022.

[17] Jacek Komorowski. Minkloc3d: Point cloud based large-scale place recognition. In *WACV*, pages 1790–1799, 2021.

[18] Boying Li, Danping Zou, Yuan Huang, Xinghan Niu, Ling Pei, and Wenxian Yu. Textslam: Visual slam with semantic planar text features. *IEEE TPAMI*, 46(1):593–610, 2023.

[19] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. In *NeurIPS*, volume 36, pages 28541–28564, 2023.

[20] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, pages 12888–12900, 2022.

[21] 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 *CVPR*, 2025.

[22] Wen Li, Yuyang Yang, Shangshu Yu, Guosheng Hu, Chenglu Wen, Ming Cheng, and Cheng Wang. Diffloc: Diffusion model for outdoor lidar localization. In *CVPR*, pages 15045–15054, 2024.

[23] Wen Li, Shangshu Yu, Cheng Wang, Guosheng Hu, Siqi Shen, and Chenglu Wen. Sgloc: Scene geometry encoding for outdoor lidar localization. In *CVPR*, pages 9286–9295, 2023.

[24] Dunqiang Liu, Shujun Huang, Wen Li, Siqi Shen, and Cheng Wang. Text to point cloud localization with multi-level negative contrastive learning. In *AAAI*, volume 39, pages 5397–5405, 2025.

[25] Son Tung Nguyen, Alejandro Fontan, Michael Milford, and Tobias Fischer. Focustune: Tuning visual localization through focus-guided sampling. In *WACV*, pages 3606–3615, 2024.

[26] Carlevaris-Bianco Nicholas, K. Ushani Arash, and M. Eustice Ryan. University of michigan north campus long-term vision and lidar dataset. *IJRR*, 35:545–565, 2015.

[27] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, volume 35, pages 27730–27744, 2022.

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, volume 32, 2019.

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, volume 30, 2017.

[30] Yanyuan Qiao, Zheng Yu, Zijia Zhao, Sihan Chen, Mingzhen Sun, Longteng Guo, Qi Wu, and Jing Liu. VL-Mamba: Exploring state space models for multimodal learning. In *NeurIPSW*, volume 262, pages 102–113, 2024.

[31] 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 *ICML*, pages 8748–8763, 2021.

[32] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regressionr. In *CVPR*, pages 3302–3312, 2019.

[33] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *CVPR*, pages 2733–2742, 2021.

[34] Yoli Shavit, Ron Ferens, and Yosi Keller. Coarse-to-fine multi-scene pose regression with transformers. *IEEE PAMI*, 2023.

[35] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[36] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *CVPR*, pages 4470–4479, 2018.

[37] Bing Wang, Chaohao Chen, Chrisxiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *AAAI*, pages 10393–10401, 2020.

[38] Fangjinhua Wang, Xudong Jiang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Glace: Global local accelerated coordinate encoding. In *CVPR*, pages 21562–21571, 2024.

[39] Shuzhe Wang, Zakaria Laskar, Iaroslav Melekhov, Xiaotian Li, Yi Zhao, Giorgos Tolias, and Juho Kannala. Hscnet++: Hierarchical scene coordinate classification and regression for visual localization with transformer. *IJCV*, 132(7):2530–2550, 2024.

[40] 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 *CVPR*, pages 5176–5185, 2023.

[41] 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*, 22:959–968, 2022.

[42] Yan Xia, Letian Shi, Zifeng Ding, Joao F Henriques, and Daniel Cremers. Text2loc: 3d point cloud localization from natural language. In *CVPR*, pages 14958–14967, 2024.

[43] 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 *CVPR*, pages 11348–11357, 2021.

[44] Kezheng Xiong, Haoen Xiang, Qingshan Xu, Chenglu Wen, Siqi Shen, Jonathan Li, and Cheng Wang. Mining and transferring feature-geometry coherence for unsupervised point cloud registration. In *NeurIPS*, volume 37, pages 35468–35491, 2024.

[45] Bochun Yang, Zijun Li, Wen Li, Zhipeng Cai, Chenglu Wen, Yu Zang, Matthias Muller, and Cheng Wang. Lisa: Lidar localization with semantic awareness. In *CVPR*, pages 15271–15280, 2024.

[46] Shangshu Yu and et al. Lidar-based localization using universal encoding and memory-aware regression. *Pattern Recognition*, 128:108685, 2022.

[47] Shangshu Yu, Xiaotian Sun, Wen Li, Chenglu Wen, Yunuo Yang, Bailu Si, Guosheng Hu, and Cheng Wang. Nidaloc: Neurobiologically inspired deep lidar localization. *IEEE TITS*, 25(5):4278–4289, 2024.

[48] Shangshu Yu, Cheng Wang, Yitai Lin, Chenglu Wen, Ming Cheng, and Guosheng Hu. Stcloc: Deep lidar localization with spatio-temporal constraints. *IEEE TITS*, 24(1):489–500, 2023.

[49] Yongzhe Yuan, Yue Wu, Xiaolong Fan, Maoguo Gong, Qiguang Miao, and Wenping Ma. Where precision meets efficiency: Transformation diffusion model for point cloud registration. In *AAAI*, volume 39, pages 9734–9742, 2025.

[50] Weiyi Zhang, Yushi Guo, Liting Niu, Peijun Li, Chun Zhang, Zeyu Wan, Jiaxiang Yan, Fasih Ud Din Farrukh, and Debing Zhang. Lp-slam: Language-perceptive rgb-d slam system based on large language model. *Complex & Intelligent Systems*, 10:5391–5409, 2024.

[51] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, pages 16816–16825, 2022.

[52] Qunjie Zhou, Sérgio Agostinho, Aljoša Ošep, and Laura Leal-Taixé. Is geometry enough for matching in visual localization? In *ECCV*, pages 407–425, 2022.

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: The abstract and introduction clearly reflect the contributions and scope of this paper.

   Guidelines:

   - The answer NA 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 NA 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: In the Appendix, we discuss the limitations of our work.

   Guidelines:

   - The answer NA 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: [NA]

Justification: Our work does not include theoretical results.

Guidelines:

- The answer NA 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: We have fully disclosed all the information needed to reproduce the experimental results of this paper.

   Guidelines:

   - The answer NA 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: The code will be open-sourced upon acceptance of the paper. We also attach the code for our implementations in the supplemental material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The main details are shown in Sec. 4.1.

   Guidelines:

   - The answer NA 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: [No]

   Justification: Following common practice among mainstream methods in the GTR-Loc field, we do not report error bars.

   Guidelines:

   - The answer NA 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: Details of the computational resources used by our model are provided in the Appendix.

   Guidelines:

   - The answer NA 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: Our work conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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: [NA]

    Justification: There is no societal impact of this work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA 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: We cite the original papers that produce the codes and datasets.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: Our paper does not release a new dataset; our code will be made available upon acceptance.

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components. We only use LLMs for language polishing.

Guidelines:

- The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

A More Experimental Results

More Dataset Details. Regarding the datasets, ground truth poses for the Oxford [1] and QEOxford [3] datasets are obtained through interpolation from an integrated GPS/INS system. For the NCLT [4] dataset, ground truth poses are generated post-collection using SLAM. Further details on the data splits can be found in Tab. 1 and Tab. 2.

More Ablation Study. Importantly, our deployed model (a distilled, LiDAR-only GTR-Loc) requires no text during inference and uses only a single LiDAR scan. However, we further investigate the case where text is used during inference. First, we train an auxiliary classification network to predict district and direction categories at test time. This network is designed with a LightLoc [2] encoder, followed by four fully-connected layers of identical feature dimensions, to separately predict partitions for position and orientation. The training configurations are consistent with those used for GTR-Loc. During inference, this classification network first predicts districts z and directions d. This prediction is then used to generate a conditioned textual input for our primary localization network. The detailed localization results alongside the position/orientation classification accuracy of this auxiliary network are presented in Tab. 3 and Tab. 4. The results (row 2) indicate that this approach delivers virtually no performance gains, while it increases both test time and computational complexity owing to the additional classification network and the LiDAR-text regression. The limited classification accuracy in the specific dataset, i.e., NCLT, produces faulty text outputs, which in turn propagate errors into the localization results. We then test a hypothetical scenario where ground truth poses are available for text generation at inference. This better result (row 3) establishes the theoretical upper bound for performance achievable with text. However, it is impractical, as gt poses are unavailable during real-world inference. Therefore, using MRD distillation is a more viable solution.

Table 1: Details of the Oxford dataset.

Sequence	Length (km)	Weather	Split
11-14-02-26	9.37	sunny	Train
14-12-05-52	9.22	overcast	Train
14-14-48-55	9.05	overcast	Train
18-15-20-12	9.04	overcast	Train
15-13-06-37	8.85	overcast	Eval
17-13-26-39	9.02	sunny	Eval
17-14-03-00	9.02	sunny	Eval
18-14-14-42	9.04	overcast	Eval

Table 2: Details of the NCLT dataset.

Sequence	Length (km)	Weather	Split
2012-01-22	6.10	overcast	Train
2012-02-02	6.20	sunny	Train
2012-02-18	6.20	sunny	Train
2012-05-11	6.00	sunny	Train
2012-02-12	5.80	sunny	Eval
2012-02-19	6.20	overcast	Eval
2012-03-31	6.00	overcast	Eval
2012-05-26	6.30	sunny	Eval

Table 3: Ablation of using text at inference.

Method	QEOxford	Oxford	NCLT
vanilla	0.83/1.12	2.67/1.25	1.46/2.80
w pred text	0.75/1.02	2.60/1.21	1.41/2.70
w gt text	0.74/0.93	2.58/1.07	1.12/2.40
ours	0.75/1.03	2.59/1.19	1.40/2.62

Table 4: Classification accuracy of position/orientation on different datasets.

Dataset	Accuracy
QEOxford	99.19%/97.64%
Oxford	98.99%/97.52%
NCLT	98.44%/85.50%

More visualization. To further dissect the performance on the Oxford and NCLT datasets, Fig. 1 and Fig. 2 provide trajectory visualizations and cumulative error distribution curves. The results of sequences 17-13-26-39 (Oxford) and 2012-02-19 (NCLT) are provided for comparison, respectively. GTR-Loc's estimated trajectory adheres closely to the ground truth throughout different sequences. Even within structurally complex or repetitive areas (as marked by blue boxes), GTR-Loc maintains consistent localization. Methods like SGLoc or LiSA exhibit jumps in these areas. The cumulative error distribution curves also demonstrate GTR-Loc's leading performance across most error ranges. Our curves for both position and orientation errors lie predominantly above those of other methods, suggesting overall lower error magnitudes.

Localization Failure Cases. The model's failures are confined to rare, extreme scenarios where LiDAR data contain almost no distinctive geometric features. For example, on a very long street in

the Oxford dataset, the building facades, streetlights, and other structures are completely identical. In these few instances (<1% of the trajectory), the teacher model succeeds by leveraging geospatial text, an external cue that the student lacks. While these specific failures can produce large errors, their statistical impact becomes negligible when averaged over the entire dataset. In the vast majority of cases, the student effectively utilizes subtle geometric cues, achieving performance nearly identical to the teacher's.

B Limitations and Future Work

Limitation. Despite its promising results, GTR-Loc has limitations that highlight areas for future research. While our distillation approach effectively eliminates the need for text processing during inference, a limitation is that leveraging text directly at inference time demonstrably achieves better performance. Our experiments demonstrate that using ground-truth text, rather than erroneously predicted text, is more beneficial for accurate localization.

Future Work. Consequently, our future work will concentrate on exploring methods for generating more accurate textual descriptions without using ground truth poses during the inference phase. The aim is to significantly enhance LiDAR-text localization precision by effectively harnessing these improved textual cues at the point of decision-making.

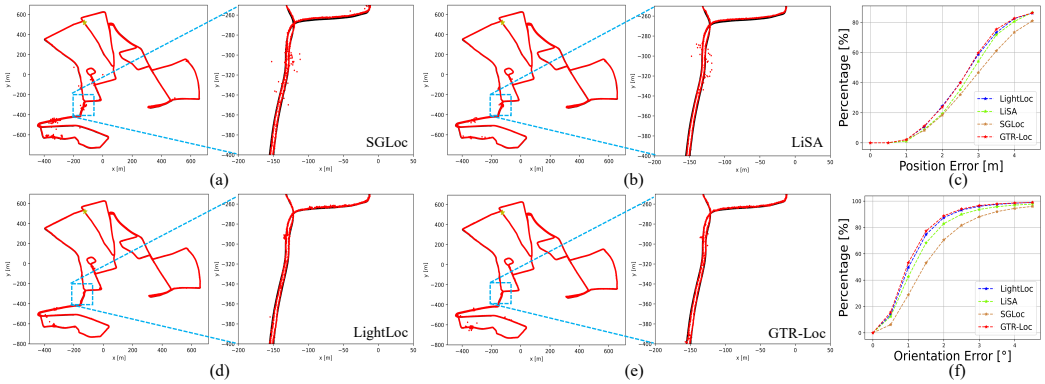

Figure 1: Visual comparisons on Oxford. (a) (b) (d) (e): predicted trajectories (red) overlaid on ground truth (black); a star marks the starting position, and the blue box highlights a challenging road segment. (c) (f): cumulative error distribution curves for position (top) and orientation (bottom).

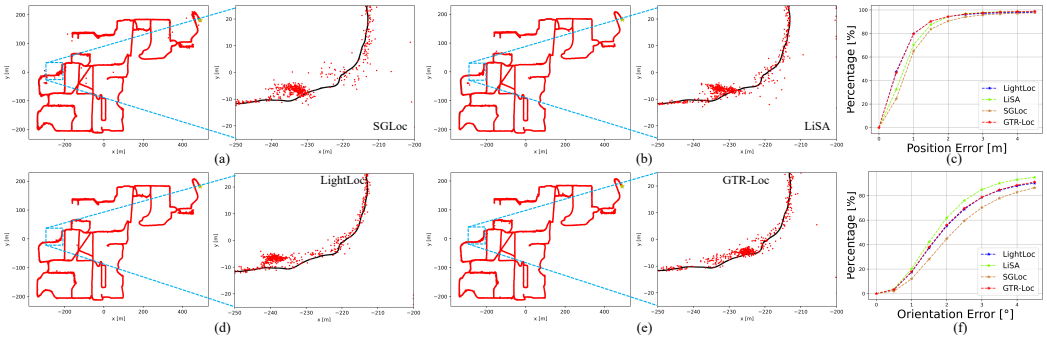

Figure 2: Visual comparisons on NCLT. The notations follow Fig. 1.

References

[1] 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 *ICRA*, pages 6433–6438, 2020.

[2] 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 *CVPR*, 2025.

[3] Wen Li, Shangshu Yu, Cheng Wang, Guosheng Hu, Siqi Shen, and Chenglu Wen. Sgloc: Scene geometry encoding for outdoor lidar localization. In *CVPR*, pages 9286–9295, 2023.

[4] Carlevaris-Bianco Nicholas, K. Ushani Arash, and M. Eustice Ryan. University of michigan north campus long-term vision and lidar dataset. *IJRR*, 35:545–565, 2015.