VectorReLoc: Reliable Vectorized SD Map Visual Re-localization with Contrastive Feature Alignment

Ziming Liu[1]✉, Quanjie Xiang[1,3], Yun Wang[2‡], Yiting Wang[2], Wen Chao[2], Zhuanjian Xu[2‡], Leichen Wang[1‡], Hao Sun[1†], and Guangyu Gao[3✉]

[1] Bosch Research, Shanghai, P.R.China
liuziming.email@gmail.com
[2] Bosch Cross-domain Computing Solutions(XC), Shanghai, P.R.China
{firstname}.{lastname}@cn.bosch.com
[3] Beijing Institute of Technology, Beijing, P.R.China
guangyugao@bit.edu.cn

Abstract. Standard-definition (SD) maps are widely used in autonomous driving, but accurate ego-centric SD map retrieval typically relies on costly RTK-GNSS joint localization. In contrast, low-cost GNSS introduces meter-level 3-DoF pose offsets with high variance, yielding misaligned local maps that can degrade downstream planning and control. Prior visual re-localization methods mostly rely on dense rasterized BEV representations, which are computation-heavy and easily distracted by scene content irrelevant to road geometry. We propose **VectorReLoc**, the first sparse re-localization framework that directly aligns vectorized SD maps with online-constructed vectorized visual maps to estimate the 3-DoF pose offset. VectorReLoc further introduces a feature alignment objective that structures the embedding space for accurate offset regression, together with a reliability predictor to identify potentially unreliable corrections and mitigate silent failures. To enable realistic training and evaluation, we additionally present a large-scale dataset with paired RTK-retrieved and GNSS-retrieved SD maps, providing real pose-offset labels rather than simulated noise used in prior work. Experiments on public benchmarks and the proposed dataset demonstrate improved accuracy, robustness, and efficiency over previous approaches.

Keywords: SD map · Visual re-localization · Online visual map construction

1 Introduction

Standard-definition (SD) maps are an indispensable component of modern autonomous driving systems. They compactly encode road geometry, topology, and traffic semantics, providing a globally available prior that supports large-scale navigation and planning. In practice, however, constructing an accurate ego-centric local SD map requires a precise vehicle pose, which is conventionally obtained from Real-Time Kinematic GNSS (RTK-GNSS) with centimeter-level accuracy. Yet the infrastructure and maintenance cost of RTK-GNSS make it difficult to scale to commercial deployment, motivating the

✉ Corresponding author; † Team lead; ‡ Project lead.

use of low-cost GNSS receivers that are already ubiquitous in consumer vehicles. Unfortunately, low-cost GNSS typically incurs meter-level errors, yielding high-variance 3-DoF pose offsets in the retrieved local SD map.

Such misalignment can severely compromise downstream tasks. When a retrieved SD map is offset from the true ego-centric pose, road links, lane boundaries, and traffic regulations become spatially misregistered, which may mislead planning and control and result in unsafe trajectories. This motivates SD map visual re-localization [26–28]: given onboard camera observations, the goal is to estimate the 3-DoF pose offset between the GNSS-retrieved SD map and its RTK-GNSS counterpart, and to warp the GNSS-retrieved map to RTK-level alignment accuracy.

Recent work tackles this task by rasterizing both camera observations and SD maps into dense bird's-eye-view (BEV) grids, and learning end-to-end BEV features to regress pose offsets [23, 26–28]. OrienterNet [23] pioneered pose-wise BEV matching against SD map tiles; MapLocNet [26] introduced coarse-to-fine registration with direct 3-DoF regression; SegLocNet [28] leveraged semantic BEV representations with multi-sensor fusion; and HOLO [27] reformulated it as homography estimation to inject projective constraints. Despite steady progress, these dense BEV pipelines share two structural limitations. First, dense grids inevitably encode all visible scene content—including buildings, vegetation, and sidewalks that are irrelevant to road-level localization, thereby injecting nuisance factors into feature space. Second, processing full-resolution raster tiles is computationally expensive relative to the geometric signal needed for map alignment, which hinders deployment on resource-constrained platforms.

In this paper, we argue that map alignment is inherently a geometric problem and is better addressed in vectorized map representations. We therefore propose **VectorReLoc**, the first sparse SD map visual re-localization framework that operates entirely on vectorized maps. Instead of rasterizing observations, we leverage an off-the-shelf online visual map construction network [14, 16, 18] to produce a vectorized visual map from onboard cameras in real time. This structured visual map—composed of polylines and polygons for lane centerlines, boundaries, and crossings—is encoded in the same tokenized form as the vectorized SD map using lightweight Transformer encoders. By operating on a small set of geometric tokens rather than high-resolution grids, VectorReLoc substantially reduces computation while naturally filtering out visually salient but localization-irrelevant content.

A central challenge is to embed the visual map (constructed under the true vehicle pose) and the GNSS-retrieved SD map (queried under an inaccurate pose) into a shared feature space in which their relative 3-DoF offset can be reliably regressed. While naive regression may work when offsets are large, it often fails to enforce the desired feature geometry in the more common small-offset regime: the visual-map feature should align with the RTK SD-map feature (both reflect accurate localization), while the GNSS SD-map feature should be separated from both. To explicitly impose this structure, we introduce a map feature alignment module based on a bidirectional InfoNCE objective augmented with a per-sample margin loss, jointly optimized with the regression loss to shape a well-behaved embedding space for offset regression.

Accuracy aside, the risk of silent failure is often neglected: models may output incorrect poses without flagging their unreliability, leading to uncaught errors in down-

stream planning. To make re-localization actionable under uncertainty, we propose a reliability prediction module that outputs a reliability score for each predicted correction. Downstream modules can consume this score independently to selectively accept or reject pose updates, improving system robustness in ambiguous scenarios.

Robust evaluation further requires both large-scale training data and trustworthy ground-truth offsets. Existing public benchmarks are limited in size and often rely on synthetically perturbed poses, which may not reflect real deployment noise. For example, nuScenes [1] and Argoverse2 [25] contain at most around 40k frames, providing limited diversity for learning reliable cross-modal alignment. To address this, we introduce a large-scale dataset with 170,533 frames across 6,534 scenes, about $4 \sim 6\times$ of the open-source data. Critically, unlike prior benchmarks that add artificial GNSS offset noise, this dataset contains paired real SD maps retrieved using RTK-GNSS and low-cost GNSS, providing realistic pose-offset labels and a faithful evaluation setting. Statistical analysis of this real-world dataset reveals that real GNSS pose offsets exhibit a Student-t distribution rather than the commonly assumed Gaussian noise. We formulate this finding as the *GNSS Offset Distribution Hypothesis*, providing a new empirical insight for GNSS SD map re-localization in autonomous driving.

The contributions of this paper can be summarized as follows.

- **Sparse vectorized re-localization.** We propose VectorReLoc, the first SD map visual re-localization method that bypasses dense BEV grids and operates directly on vectorized map representations, achieving the best accuracy with significantly reduced computational cost.
- **Contrastive map feature alignment.** We introduce a feature alignment method that explicitly structures the embedding space for reliable GNSS offset regression.
- **Reliability prediction.** We propose the reliability scoring module for visual re-localization, filtered reliable samples have higher recall and lower error, enabling downstream modules to detect and selectively accept reliable pose corrections.
- **New dataset and benchmarks.** We introduce a new large-scale dataset that, unlike all prior benchmarks relying on synthetically noised poses, contains paired real RTK-GNSS SD maps and GNSS SD maps, reflecting true deployment conditions. We also find that real GNSS offset noise follows a Student-*t* distribution.

2 Related Works

SD Map Visual Re-localization. Visual re-localization with SD maps aims to correct a noisy GNSS prior by aligning onboard visual observations with low-cost, globally available SD maps. OrienterNet [23] pioneered neural BEV matching against SD map tiles via pose-wise score volumes. MapLocNet [26] shifted to coarse-to-fine feature registration with direct 3-DoF offset regression for real-time operation. SegLocNet [28] strengthened the representation side with BEV semantic segmentation matched against a binary map mask, fusing camera and LiDAR inputs. HOLO [27] reformulates pose estimation as homography estimation, injecting projective constraints for faster convergence. All these methods rely on dense rasterized BEV representations, incurring substantial computation and sensitivity to irrelevant scene content. In contrast, our method

operates directly on sparse vectorized SD and visual maps, enabling map feature alignment while discarding noisy visual elements.

Online Visual Map Construction. Online visual map construction has evolved from raster BEV segmentation toward end-to-end vectorized element generation with Transformer decoders. MapTR [16] first formulated each map element as a permutation-equivalent set prediction problem in BEV; MapTRv2 [17] refined this with auxiliary dense supervision and decoupled attention; MapQR [18] further improved instance coherence via scatter-and-gather queries. LaneSegNet [14] organizes road structure around lane segments for joint geometry and connectivity prediction. SMERF [20] shows that SD map priors can regularize online mapping. These structured vector map outputs serve as the direct input representation for our re-localization framework.

Contrastive Learning. Contrastive learning optimizes feature representations by maximizing agreement between positive pairs while repelling negatives under an InfoNCE objective [21]. Representative frameworks such as MoCo [8], SimCLR [3], and SupCon [11] have demonstrated its effectiveness in structuring discriminative embedding spaces. In this work, we introduce a bidirectional InfoNCE formulation augmented with a per-sample margin loss to explicitly enforce the desired geometric structure for map alignment: visual map features are pulled toward RTK SD map features, while GNSS SD map features are pushed away from both. To our knowledge, contrastive learning has not been explored for vectorized SD map feature alignment.

Reliability in Autonomous Driving. Reliable autonomy requires models to signal when their predictions are untrustworthy, as silent failures can propagate to downstream tasks. Prior work addresses this via uncertainty estimation [4, 10, 12], confidence calibration [5, 7], and OOD/failure detection [9, 13, 15]. These methods largely target classification; applying them to continuous offset regression for SD map re-localization is non-trivial, as incorrect alignments can appear plausible and failure labels are unavailable. We therefore propose a per-dimension confidence scoring module for visual re-localization, requiring no annotated failure cases.

3 Method

We present **VectorReLoc**, a sparse SD map visual re-localization framework operating on vectorized map representations instead of dense rasters. An online mapping network produces a vectorized visual map from multi-camera inputs, which is encoded alongside the GNSS-localized SD map by modality-specific Transformer encoders. A contrastive feature alignment mechanism structures the embedding space so that a lightweight MLP can reliably regress the 3-DoF pose offset $(\Delta x, \Delta y, \Delta\theta)$, while a reliability prediction module flags potentially unreliable corrections. Fig. 1 illustrates the pipeline.

3.1 Online Map Construction Module

VectorReLoc uses a vectorized visual map produced at runtime by an off-the-shelf BEV detection Transformer such as MapTR [16], MapQR [18], or LaneSegNet [14]. These networks lift image features to BEV and decode lane-level polylines and polygons via a detection Transformer, yielding a vectorized visual map $\mathscr{M}_{vis}$ of lane centerlines,

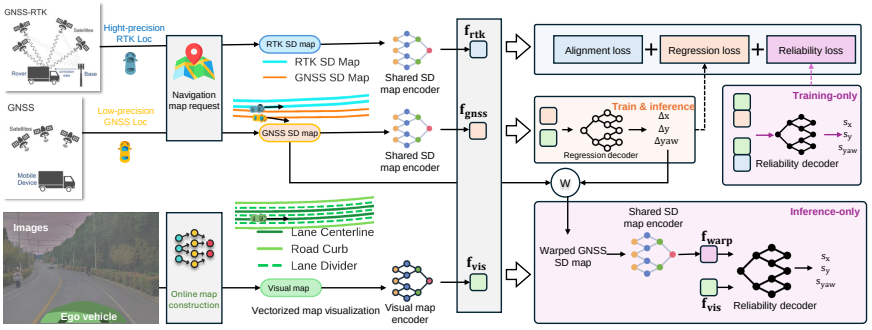


Fig. 1: Overall structure of VectorReLoc. Components are color-coded: blue denotes the RTK SD map (used exclusively for feature alignment loss during training); orange represents the GNSS SD map. Note that the SD map encoders share parameters. Green signifies the visual map derived from camera images. During training, the pretrained online map construction network remains frozen. While the regression decoder is consistent across stages, the reliability decoder utilizes different inputs for training versus inference.

boundaries, and crossings in ego-centric coordinates. This structured output naturally focuses on road geometry and discards irrelevant visual content. This online map construction network is pre-trained with L1 position loss and cross-entropy classification loss following [14, 16, 18]. Its parameters are frozen during VectorReLoc training.

3.2 Map Encoder

SD maps are lightweight navigation maps that encode only road-level geometry and semantics. We augment their geometry into lane-level vectors using given lane number and a fixed 3 m lane width, so that both SD and visual maps share a consistent vectorized form. Each vectorized map is a set of N elements (polylines or polygons), parameterized by sampled point coordinates and element type. These N tokens are contextualized via stacked multi-head self-attention (MHSA) layers [24]: $\mathbf{V}^{(l)} = \mathrm{MHSA}\left(\mathbf{V}^{(l-1)}\right), \quad l = 1, \ldots, L$, where $\mathbf{V}^{(0)} \in \mathbb{R}^{N \times d}$ is the initial embedding and L the layer count. A learnable global query $\mathbf{q} \in \mathbb{R}^{1 \times d}$ then aggregates all contextualized tokens via cross-attention: $f = \mathrm{CrossAttn}\left(\mathbf{q}, \mathbf{V}^{(L)}\right) \in \mathbb{R}^{d}$, producing a fixed-size global map feature f. Because SD maps and visual maps have structurally different element types, they use two independent encoders. Within the SD map branch, SD maps $\mathscr{M}_{rtk}$ and $\mathscr{M}_{gnss}$ share a weight-tied encoder so that f_{rtk} and f_{gnss} are directly comparable; the visual map encoder produces f_{vis} independently. Operating on only N tokens keeps cost well below that of dense BEV grids.

3.3 Map Feature Alignment with Contrastive Learning (CL)

For accurate offset regression, the feature space must satisfy: (i) f_{vis} and f_{rtk} should be close (both reflect accurate localization), and (ii) f_{gnss} with low-precision GNSS localization should be separated from both. Pure regression provides no such guarantee, so we introduce a contrastive alignment mechanism.

Pair construction. For mini-batch of size B, the positive pair for sample i is (f^i_{vis}, f^i_{rtk}). Negatives comprise (i) cross-modal in-batch features and (ii) all GNSS features $\{f^j_{gnss}\}_{j=1}^{B}$ as hard negatives, which simultaneously push f_{gnss} away from f_{vis} and f_{rtk}. A GNSS pose with x, y, and yaw all close to the RTK pose is rare, and since each positive is contrasted against $2B$ keys, such accurate GNSS negatives have limited influence.

Bidirectional InfoNCE loss. Following [21], we compute InfoNCE in both directions and average. For visual queries f_{vis} with keys $[f_{rtk}; f_{gnss}] \in \mathbb{R}^{2B\times d}$:

$$\mathscr{L}^{vis}_{\mathrm{NCE}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp\big(\mathrm{sim}(f^i_{vis}, f^i_{rtk})/\tau\big)}{\sum_{j=1}^{B}\exp\Big(\mathrm{sim}(f^i_{vis}, f^j_{rtk})/\tau\Big)+\sum_{j=1}^{B}\exp\big(\mathrm{sim}(f^i_{vis}, f^j_{gnss})/\tau\big)}, \quad (1)$$

with $\mathscr{L}^{rtk}_{\mathrm{NCE}}$ defined symmetrically (f_{rtk} as queries, $[f_{vis}; f_{gnss}]$ as keys):

$$\mathscr{L}_{\mathrm{NCE}} = \tfrac{1}{2}\Big(\mathscr{L}^{vis}_{\mathrm{NCE}} + \mathscr{L}^{rtk}_{\mathrm{NCE}}\Big), \quad (2)$$

where $\mathrm{sim}(\cdot,\cdot)$ is cosine similarity and τ is a temperature.

Push loss. InfoNCE ranks similarities within the batch but does not guarantee a fixed per-sample gap between the positive and the negatives. We add a margin-based hinge loss: with positive similarity $p_i = \mathrm{sim}(f^i_{vis}, f^i_{rtk})$ and GNSS-negative similarities $n^{vg}_i = \mathrm{sim}(f^i_{vis}, f^i_{gnss})$, $n^{rg}_i = \mathrm{sim}(f^i_{rtk}, f^i_{gnss})$, it forces p_i to exceed both by a margin m:

$$\mathscr{L}_{\mathrm{push}} = \frac{1}{B}\sum_{i=1}^{B}\big[\max\big(0,\, n^{vg}_i - p_i + m\big) + \max\big(0,\, n^{rg}_i - p_i + m\big)\big], \quad (3)$$

where $m > 0$ is a margin ensuring that f_{vis} and f_{rtk} are at least m closer to each other than either is to f_{gnss}.

$$\mathscr{L}_{\mathrm{align}} = \mathscr{L}_{\mathrm{NCE}} + \lambda_{\mathrm{push}}\mathscr{L}_{\mathrm{push}}, \quad (4)$$

where λ_{push} balances the two terms. Crucially, $\mathscr{L}_{\mathrm{align}}$ is optimized jointly with the regression loss (Sec. 3.4). Ablations confirm that joint training is key to preserving alignment during regression (Sec. 5.5).

3.4 Localization Pose Offset Regression Module

With the aligned feature space, this module simplifies to a lightweight regression on the concatenated global map features. The concatenated representation $\mathbf{z} = [f_{vis}; f_{gnss}]$ is fed to an MLP network:

$$(\Delta x, \Delta y, \Delta\theta) = \mathrm{MLP}([f_{vis}; f_{gnss}]). \quad (5)$$

This regression decoder is supervised with smooth-ℓ_1 (SL1) loss for robustness to outliers:

$$\mathscr{L}_{\mathrm{reg}} = \mathrm{SL1}(\Delta x, \Delta x^{gt}) + \mathrm{SL1}(\Delta y, \Delta y^{gt}) + \lambda_\theta\,\mathrm{SL1}(\Delta\theta, \Delta\theta^{gt}), \quad (6)$$

where λ_θ balances the translational and rotational components.

3.5 Reliability Prediction Module

While confidence-based gating has been explored for classification [5], applying it to continuous pose-offset regression for SD map re-localization is non-trivial: incorrect alignments can appear geometrically plausible and no explicit failure labels are available. We therefore introduce a per-dimension reliability predictor, trained with pseudo-labels from the ground-truth offset magnitude, that enables downstream modules to selectively accept or reject each component of the predicted offset.

Module architecture. The predictor reuses the global features f_{vis} and $f_{sd} \in \mathbb{R}^d$ from the map encoders, stacking them into $\mathbf{K} = [f_{vis}; f_{sd}] \in \mathbb{R}^{2\times d}$. Here f_{sd} denotes the SD map feature used by this module, instantiated as f_{rtk}/f_{gnss} during training and as f_{warp} at inference. A learnable score query $\mathbf{q}_s \in \mathbb{R}^d$ attends over $\mathbf{K}$ via cross-attention: $\mathbf{h}_s = \mathrm{CrossAttn}(\mathbf{q}_s,\ \mathbf{K},\ \mathbf{K}) \in \mathbb{R}^d$. An MLP network followed by sigmoid σ produces the per-dimension scores: $\hat{\mathbf{s}} = (\hat{s}_x, \hat{s}_y, \hat{s}_\theta) = \sigma(\mathrm{MLP}(\mathbf{h}_s)) \in [0,1]^3$.

Training with pseudo-label. Each sample undergoes a positive pass with $\mathbf{K}^+ = [f_{vis}; f_{rtk}]$ (target $\mathbf{s}^+ = (1,1,1)$) and a negative pass with $\mathbf{K}^- = [f_{vis}; f_{gnss}]$, whose pseudo-labels are derived from the ground-truth offset magnitude. Each dimension is normalized independently with respect to its predefined offset range (r_x, r_y, r_θ):

$$\alpha_x = \mathrm{clamp}\big(|\Delta x^{gt}|/r_x, 0, 1\big), \quad \alpha_y = \mathrm{clamp}\big(|\Delta y^{gt}|/r_y, 0, 1\big), \quad \alpha_\theta = \mathrm{clamp}\big(|\Delta \theta^{gt}|/r_\theta, 0, 1\big), \tag{7}$$

Then, negative pseudo label is $\mathbf{s}^- = (1-\alpha_x,\ 1-\alpha_y,\ 1-\alpha_\theta) \in [0,1]^3$. A small offset ($\alpha_k \approx 0$) yields a reliable pseudo-label near 1. The reliability scores from both passes are supervised jointly with binary cross-entropy (**BCE**), summed over dimensions:

$$\mathscr{L}_{\mathrm{rel}} = -\frac{1}{2B}\sum_{i=1}^{B}\sum_{k\in\{x,y,\theta\}}\Big[\log \hat{s}^+_{i,k} + s^-_{i,k}\log \hat{s}^-_{i,k} + (1-s^-_{i,k})\log(1-\hat{s}^-_{i,k})\Big]. \tag{8}$$

All components are optimized jointly: $\mathscr{L} = \mathscr{L}_{\mathrm{reg}} + \lambda_{\mathrm{align}}\mathscr{L}_{\mathrm{align}} + \lambda_{\mathrm{rel}}\mathscr{L}_{\mathrm{rel}}$, where λ_{align} and λ_{rel} are balancing weights.

Inference. At test time, GNSS SD map is firstly warped with the predicted offset 3-DoF correction pose, where each map vertex $\mathbf{p}$ is transformed by the predicted SE(2) offset as $\hat{\mathbf{p}} = R(\Delta\theta)\mathbf{p} + (\Delta x, \Delta y)^\top$. The warped SD map $\mathscr{M}_{warp}$ is re-encoded to obtain $f_{warp} \in \mathbb{R}^d$. Finally, the paired key $\mathbf{K} = [f_{vis}; f_{warp}]$ is passed to this network, generating reliability scores that downstream modules can consume independently.

4 Datasets

4.1 New Benchmark: VectorReLoc Dataset

To address the lack of large-scale, diverse benchmarks for SD map visual re-localization, this paper introduces a large-scale dataset, namely **VectorReLoc Dataset**.

Table 1: Comparison of datasets used in SD map visual re-localization.

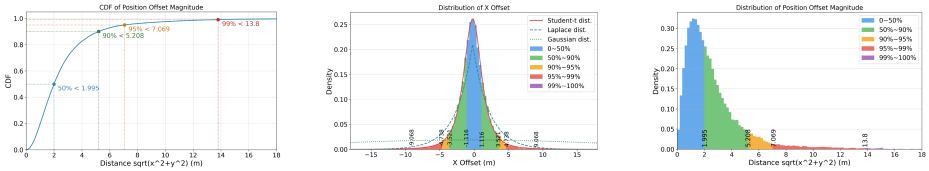
Dataset	Scenes	Frames	Freq.	RTK-GNSS-Sensor Loc.	GNSS-Sensor Loc.
nuScenes [1]	~1,000	40,157	2 Hz	✗	✓
Argoverse2 [25]	~850	27,000	2 Hz	✗	✓
VectorReLoc(Ours)	**6,534**	**170,533**	1 Hz	✓	✓

Dataset Overall. The dataset comprises approximately 6,534 scenes with 170,533 frames captured at 1 Hz, covering a wide variety of geographic regions, road topologies, and environmental conditions (e.g., urban, suburban, highway, night, rain). Each frame provides 8 surround-view images at 1024×768, low-cost GNSS/IMU measurements together with dual RTK poses, and the corresponding SD maps retrieved from a high-quality commercial map provider. The data are collected across Shanghai, Jiangsu, and Anhui, China. This scale is significantly larger than all existing open-source alternatives, enabling more reliable training and evaluation of data-hungry deep learning approaches. There are 135,735 training samples with 5,228 scenes and 34,798 validation samples with 1,306 scenes. Table 1 summarizes a comparison with existing datasets.

Data Annotation. VectorReLoc dataset provides real-world GNSS SD map, RTK SD map, as well as the translational and rotational ground-truth offsets between the GNSS pose and the RTK pose. All sequences are recorded with dual u-blox A9 receivers and an SMI980 IMU, where the coarse pose is produced by our VMPS INS-fusion and the high-accuracy pose by two different RTK algorithms. We conduct a comprehensive statistical analysis of these offsets across all 170,533 frames in the VectorReLoc dataset. Table 2 summarizes the detailed statistics.

Table 2: Statistics of GNSS pose offsets in VectorReLoc dataset. x and y denote lateral and longitudinal translational offsets, "Position" is Euclidean distance, and "Yaw" is heading offset.

Offset	Mean	Min	Max	50%	90%	95%	99%
x (m)	−0.96	−786.43	166.07	1.12	3.52	4.74	9.07
y (m)	−0.19	−188.33	146.20	1.15	3.85	5.44	11.93
2D Position (m)	3.45	0.007	792.95	2.00	5.21	7.07	13.80
Yaw (deg)	0.04	−56.17	14.13	0.09	0.33	0.51	1.40



(a) Position distance CDF **(b)** x offset histogram **(c)** Position distance histogram

(d) Yaw offset CDF **(e)** y offset histogram **(f)** Yaw offset histogram

Fig. 2: Empirical offset distributions on VectorReLoc. **CDF**: cumulative distribution function.

As shown in Fig. 2, all three offset components exhibit extremely heavy-tailed behavior, with excess kurtosis values of 878, 466, and 2324 for x, y, and yaw. Among Gaussian, Laplace, and Student-t fits evaluated by AIC on all 170,533 samples, the Student-t is the clear winner, with near-zero KS statistics and both Gaussian and Laplace firmly rejected. The fitted parameters have low degrees of freedom: $x \sim t(\nu{=}2.44,$

μ=−0.21, s=1.39), $y \sim t(\nu$=2.24, μ=0.10, s=1.44), and yaw $\sim t(\nu$=1.56, μ=0.03°, s=0.09°). This indicates heavy-tailed offsets: x and y retain finite variance but have undefined higher-order moments, while yaw has infinite theoretical variance because ν<2. The 2D position offset has a median of 2.00 m with 90% of frames within 5.21 m, yet tails extend to 792.95 m. This heavy-tailed real offsets contrasts with the previous Gaussian simulation protocol, motivating a robust loss and a reliability-aware module.

4.2 Open-source Datasets

Following prior works [26–28], we benchmark our method on nuScenes and Argoverse2 using original sensor imagery. For Argoverse2, vectorized SD maps are retrieved from real OpenStreetMap [22], identical to prior works. For nuScenes, its inaccurate GPS makes the raw OSM retrieval severely misaligned and yields wrong ground-truth labels for all methods; we therefore construct RTK-aligned pseudo SD maps from the HD map as a clean reference, and re-train MapLocNet [26] under the same pseudo map for a fair comparison. While the KITTI dataset [6] provides RTK-level precision, it currently lacks map annotations. We leave the construction of a KITTI-based benchmark for future work.

5 Experiments

We first compare with state-of-the-art methods using standard simulated-offset protocols. Subsequently, we demonstrate the benefit of our proposed large-scale dataset through pretraining experiments. We further analyze the gap between simulated and real GNSS offsets, and ablate core modules: contrastive feature alignment and reliability prediction.

5.1 Implementation Details

Metrics. Following [26, 28], we report *recall at threshold.*, which is the fraction of frames whose prediction error is below a threshold τ:

$$\mathrm{PR}@\tau_{\mathrm{pos}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\left[\sqrt{(\Delta x_i)^2+(\Delta y_i)^2} \le \tau_{\mathrm{pos}}\right], \quad \mathrm{OR}@\tau_{\mathrm{ori}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}[|\Delta\theta_i| \le \tau_{\mathrm{ori}}], \tag{9}$$

where $\Delta x_i, \Delta y_i$ are the lateral and longitudinal errors and $\Delta\theta_i$ the heading error (wrapped to $(-180°, 180°]$). We report PR at $\{1,2,5,10\}$ m and OR at $\{1°, 2°, 5°, 10°\}$.

We additionally report mean absolute errors per axis and jointly:

$$E_x = \frac{1}{N}\sum_i |\Delta x_i|,\ E_y = \frac{1}{N}\sum_i |\Delta y_i|,\ E_{pos} = \frac{1}{N}\sum_i \sqrt{\Delta x_i^2+\Delta y_i^2},\ E_{ori} = \frac{1}{N}\sum_i |\Delta\theta_i|. \tag{10}$$

E_x, E_y, E_{pos} are in meters; E_{ori} in degrees (↓). Separating E_x and E_y captures direction-dependent sensitivities to road geometry and GNSS noise.

Architecture. Both map encoders use d=256, 8 attention heads, and 10 Transformer layers. Each polyline is sampled to N_p=20 points with sinusoidal positional embeddings (dim=32). The perception range is 200×100 m ($x\in[-100, 100]$, $y\in[-50, 50]$ m) centred on the ego vehicle.

Training. The model is trained end-to-end for 100 epochs with AdamW [19] (lr 10^{-4}, weight decay 0.01), cosine annealing (500-step warm-up, ratio $1/3$, min lr 10^{-7}), and gradient clipping at $\|\cdot\|_2$=35. The contrastive loss uses temperature τ=0.07 and weight 1.0; the smooth-ℓ_1 regression loss has weight 1 for $(\Delta x, \Delta y)$ and 100 for $\Delta\theta$ (in radians). Training runs on 8×A100 GPUs with per-GPU batch size 48 (total 384).

Inference. The model runs in single-sample mode without data augmentation.

5.2 Comparison with Previous Methods on Open-source Data

We compare the re-localization recall of our method against previous state-of-the-art approaches on the open-source Argoverse2 and nuScenes dataset. The results are summarized in Tab. 3 and Tab. 4.

Table 3: Quantitative results on the nuScenes dataset. Performance is evaluated under a simulation configuration with initial offsets of ±30 m for x,y coordinates and ±30° for yaw. SD map: Standard-Definition map; Cam: Camera; Lid: LiDAR. The best-performing results are highlighted in **bold**.

Method	Input	Position Recall (%) ↑				Orientation Recall (%) ↑				$E_{pos}(m)\downarrow$	E_{ori} (°) ↓
		1 m	2 m	5 m	10 m	1°	2°	5°	10°		
OrienterNet [23]	SDmap+Cam	15.74	33.27	49.86	60.10	31.69	45.81	61.23	72.44	15.47	26.43
U-BEV [2]	SDmap+Cam	16.89	41.60	71.33	83.46	–	–	–	–	–	–
MapLocNet [26]$^{\texttt{paper}}$	SDmap+Cam	20.10	45.54	77.70	91.89	58.61	84.10	96.23	98.62	–	–
MapLocNet [26]$^{\texttt{reproduced}}$	SDmap+Cam	22.15	48.71	78.60	93.24	56.89	82.69	96.54	99.32	3.45	1.32
SegLocNet [28]	SDmap+Cam+Lid	35.63	57.98	74.55	80.09	38.58	64.98	83.62	87.40	8.15	19.68
HOLO [27]	SDmap+Cam	30.40	54.36	82.11	92.46	69.76	88.33	**98.14**	99.18	3.76	1.06
Ours	SDmap+Cam	**65.79**	**83.77**	**95.02**	**98.77**	**87.64**	**93.55**	97.84	**99.20**	**1.36**	**0.62**

Table 4: Results on Argoverse2 dataset. Evaluation performed under configurations with initial offsets of ±30 m for x,y coordinates and ±30° for yaw. Best results are highlighted in **bold**.

Method	Position Recall (%) ↑				Orientation Recall (%) ↑				$E_{pos}(m)\downarrow$	E_{ori} (°) ↓
	1 m	2 m	5 m	10 m	1°	2°	5°	10°		
(2024)MapLocNet [26]	23.26	47.24	79.13	94.33	62.35	86.28	**96.24**	98.61	-	-
(2025)SegLocNet-SD [28]	40.72	67.99	76.98	80.86	42.96	71.52	80.38	81.42	8.66	34.03
(2026)HOLO [27]	29.38	51.89	77.26	-	65.61	86.33	95.86	-	-	-
Ours	**58.05**	**81.80**	**95.47**	**99.12**	**75.08**	**89.19**	95.93	**98.71**	**1.42**	**1.06**

VectorReLoc consistently outperforms all prior methods on both benchmarks. On **nuScenes** (Tab. 3), VectorReLoc achieves a 1 m Position Recall of **65.79%** (+30.16 pp over SegLocNet [28], which uses additional LiDAR), reducing E_{pos} from 3.76 m (HOLO) to **1.36 m** and E_{ori} from 1.06° to **0.62**°. On **Argoverse2** (Tab. 4), this model reaches a 1 m recall of **58.05%** (+17.33 pp over SegLocNet), with E_{pos} and E_{ori} reduced by **83.6%** and **96.9%** respectively compared to SegLocNet. The only exception is the 5° orientation threshold on AV2, where MapLocNet [26] marginally leads (96.24% vs. 95.93%). Our method surpasses it on all other metrics. This 5° threshold is of limited practical relevance, as 99% of real frames have $|\Delta\theta| < 1.40°$, shown in Tab. 2. And on the stricter 1°/2° orientation recalls our method leads by a large margin.

5.3 Improvement with the Model Pretraining on Proposed Dataset

To investigate whether our large-scale dataset provides transferable representations, we pretrain VectorReLoc on it and fine-tune on the Argoverse2 and nuScenes benchmarks. For each benchmark we compare training from scratch against initialization from the pretrained model and fine-tuning end-to-end, using the LaneSegNet [14] and MapQR [18] as the online map construction module on Argoverse2 and nuScenes separately.

Table 5: Ablation study of pretraining.

Pretrained on proposed VectorReLoc dataset and finetuning on Argoverse2.

Method	Position Recall (%) ↑				Orientation Recall (%) ↑				$E_{pos}(m)\downarrow$	E_{ori} (°) ↓
	1 m	2 m	5 m	10 m	1°	2°	5°	10°		
from scratch	49.86	76.83	93.26	97.87	64.52	83.11	95.14	98.37	1.76	1.35
with Pretraining	**58.05**	**81.80**	**95.47**	**99.12**	**75.08**	**89.19**	**95.93**	**98.71**	**1.42**	**1.06**

Pretrained on the proposed VectorReLoc dataset and finetuning on nuScenes.

Method	Position Recall (%) ↑				Orientation Recall (%) ↑				$E_{pos}(m)\downarrow$	E_{ori} (°) ↓
	1 m	2 m	5 m	10 m	1°	2°	5°	10°		
from scratch	39.09	65.48	90.85	98.09	82.69	91.78	97.59	99.29	2.15	0.77
with Pretraining	**65.79**	**83.77**	**95.02**	**98.77**	**87.64**	**93.55**	**97.84**	**99.20**	**1.36**	**0.62**

Tab. 5 summarises the results. Pretraining on our large-scale dataset consistently improves performance across all metrics on both benchmarks, demonstrating strong cross-dataset transfer. On **Argoverse2**, the 1 m Position Recall improves from 49.86% to **58.05%** (+8.19 pp, +16.4% relative) and the 1° Orientation Recall from 64.52% to **75.08%** (+10.56 pp, +16.4% relative), with E_{pos} reduced from 1.76 m to **1.42 m** (-0.34 m, -19.3%) and E_{ori} from 1.35° to **1.06**° ($-0.29°$, -21.5%). On **nuScenes**, gains are even more pronounced: the 1 m recall rises from 39.09% to **65.79%** (+26.70 pp, +68.3% relative), the 1° Orientation Recall from 82.69% to **87.64%** (+4.95 pp), E_{pos} from 2.15 m to **1.36 m** (-0.79 m, -36.7%), and E_{ori} from 0.77° to **0.62**° ($-0.15°$, -19.5%). These improvements confirm that pre-exposure to diverse road geometries in the large-scale data enables the map encoders to learn more generalisable structural representations, benefiting fine-tuning on unseen benchmarks.

5.4 GNSS Offset Distribution Hypothesis

All prior works [23,26–28] inject synthetic pose offsets for training, yet none has established what distribution real GNSS offsets follow. Open-source benchmarks typically simulate up to 30 m, whereas Tab. 2 shows 90% of real offsets in VectorReLoc fall within 5.21 m. We characterise the true distribution using our 170,533 paired GNSS–RTK frames and validate it with two experiments.

Exp. 1: Validating the distribution hypothesis (Tab. 6). We evaluate all four train/test combinations of offset source: **S**ynthetic (Student-t distributions fitted to Tab. 2) and **R**eal (real GNSS). On each test split, training on S vs. R makes no measurable difference. S→S vs. R→S and S→R vs. R→R each differ by $\leq$0.01 m in E_{pos} and $\leq$0.07 pp in 5 m recall. This result confirms that real GNSS offsets are fully captured by independent Student-t distributions. Based on this evidence, we formulate the following

Table 6: Cross-evaluation of synthetic (**S**, Student-t distributions fitted from real data) vs. real (**R**) GNSS offsets on the VectorReLoc test split. Best in **bold**.

Train	Test	Error ↓			Position Recall (%) ↑			Abs. Error ↓	
		xy (m)	x (m)	y (m)	1 m	2 m	5 m	E_{pos} (m)	E_{ori} (°)
Synthetic	Synthetic (S→S)	**2.80**	**1.64**	**1.92**	**10.45**	**33.44**	**91.24**	**3.19**	**4.54**
Real	Synthetic (R→S)	2.81	1.64	1.93	10.44	33.20	91.21	3.19	4.72
Synthetic	Real (S→R)	2.21	**1.37**	1.44	**21.92**	**54.72**	**93.18**	**2.68**	7.00
Real	Real (R→R)	**2.21**	1.38	**1.44**	21.55	54.70	93.11	2.68	**6.73**

empirical **GNSS Offset Distribution Hypothesis**:

$$x \sim t(\nu_x, \mu_x, s_x), \quad y \sim t(\nu_y, \mu_y, s_y), \quad \theta \sim t(\nu_\theta, \mu_\theta, s_\theta), \tag{11}$$

where ν, μ, and s denote degrees of freedom, location, and scale of each axis, and the 2D distance $d=\sqrt{x^2+y^2}$ is right-skewed with a heavy tail (Fig. 2 (c)). We hypothesize that the Student-t distributional family generalizes across GNSS hardware and environments, with only the nine parameters varying.

Table 7: Performance with simulated offset scale on Argoverse2. Best in **bold**.

Offset r	Position Recall (%) ↑				Orientation Recall (%) ↑				E_{pos} (m) ↓	E_{ori} (°) ↓
	1 m	2 m	5 m	10 m	1°	2°	5°	10°		
1 m	54.00	93.33	**100.00**	**100.00**	**82.67**	**93.33**	**99.67**	**100.00**	1.04	0.67
3 m	**68.02**	**87.66**	98.73	**100.00**	78.13	93.32	98.75	99.96	**1.01**	0.75
5 m	64.54	86.31	97.39	99.87	77.77	91.42	98.50	99.83	1.14	0.80
7 m	59.65	84.18	95.93	99.23	76.00	91.48	98.25	99.83	1.34	0.84
9 m	57.17	83.09	96.35	98.87	72.24	89.67	97.98	99.71	1.39	0.93
11 m	54.62	82.28	95.37	98.52	72.99	89.84	98.16	99.77	1.51	0.92
13 m	49.11	78.17	93.95	98.02	70.05	88.39	97.68	99.62	1.74	0.99
30 m	50.24	78.46	94.61	98.00	70.40	87.98	97.04	98.91	1.67	1.09

Exp. 2: Impact of the simulation-scale gap (Tab. 7). Eq. (11) reveals an offset scale gap between open-source benchmark (99% x,y offset < 30 m, 30 m) and reality (99% x,y offset < 9.07 m, 11.93 m). We train on open-source Argoverse2 with simulation ranges $r \in \{1,3,5,7,9,11,13,30\}$ m to assess the impact of x and y offset scale. Heading angle offsets are sampled within ± 0.1 rad. For 1m $\rightarrow$ 30m in offset scale, 5 m position recall stays above 93% and E_{pos} rises only from 1.04 m to 1.74 m, showing graceful degradation. Therefore, this scale gap does not invalidate existing open-source benchmark results.

5.5 Map Feature Alignment with Contrastive Learning

We ablate the map feature alignment across three training configurations: (i) **Regression-only**: direct offset regression with no contrastive objective; (ii) **CL Pretraining + Fine-tune**: contrastive pre-training of the map encoders followed by regression fine-tuning; and (iii) **CL Joint Training**: simultaneous optimization of the contrastive and regression objectives. All variants share a same network structure on VectorReLoc dataset.

As shown in Table 8, CL Joint Training outperforms both baselines on most metrics, especially the translational metrics. Under real GNSS noise, it reduces the absolute

Table 8: Ablation study on map feature alignment on the VectorReLoc dataset. Results are reported under (a) simulated GNSS noise ($3\sigma_{x,y}$=5 m, $3\sigma_{\theta}$=1°) and (b) real GNSS noise.

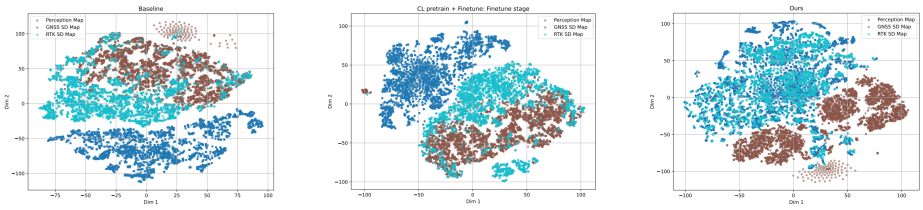
(a) Simulated GNSS noise

Method	Error ↓				Position Recall (%) ↑				Orientation Recall (%) ↑			
	xy (m)	*x* (m)	*y* (m)	yaw (°)	1 m	2 m	5 m	10 m	1°	2°	5°	10°
Regression-only (Baseline)	4.7645	4.0042	1.8068	1.0211	5.25	16.91	59.67	94.44	73.05	86.85	96.52	99.27
CL Pretraining + Finetune	3.5085	3.0356	1.1145	0.4988	17.69	39.34	76.64	95.58	89.64	96.32	**98.94**	**99.75**
CL Joint Training	**3.2324**	**2.7414**	**1.1190**	**0.4969**	**19.27**	**42.24**	**79.80**	**96.70**	**90.00**	**96.36**	98.92	99.67

(b) Real GNSS noise

Method	Error ↓				Position Recall (%) ↑				Orientation Recall (%) ↑			
	xy (m)	*x* (m)	*y* (m)	yaw (°)	1 m	2 m	5 m	10 m	1°	2°	5°	10°
Regression-only (Baseline)	3.0632	2.1082	1.6938	0.1601	22.96	53.57	90.51	97.97	98.68	99.49	99.79	99.92
CL Pretraining + Finetune	1.5250	1.2208	0.6356	0.1130	52.40	78.96	97.16	**99.45**	**99.43**	99.76	99.92	**99.98**
CL Joint Training	**1.3740**	**1.0594**	**0.6189**	**0.1037**	**57.09**	**83.01**	**97.79**	99.43	99.40	**99.77**	**99.94**	99.97

position error by **55.1%** (3.06 m → 1.37 m) and improves 1-m recall by **+34.1 pp**, confirming that contrastive alignment strengthens the spatial geometric constraints learned by map encoders. CL Pretrain + Finetune also surpasses the baseline but falls short of joint training, because regression fine-tuning partially destroys the contrastive alignment established in Stage 1 (see Fig. 3). Across all three configurations, the largest gains are on translational localization, while yaw error also improves modestly.



(a) Baseline **(b)** CL Pretraining + Finetune **(c)** Joint Training

Fig. 3: t-SNE visualization of map features extracted from the test set under each training configuration. Each point represents a global map feature vector; colors denote different map modalities: • visual map (f_{vis}) , • RTK SD map (f_{rtk}), • GNSS SD map (f_{gnss}).

Feature Distribution Analysis (Fig. 3). The t-SNE visualizations reveal the mechanism underlying the quantitative gains. In the **Baseline** (a), visual map features and SD map features occupy clearly separated regions, while both RTK SD map and GNSS SD map heavily entangled—regression training alone can not provide cross-modal bridge RTK SD map and visual map. In **CL Pretraining + Finetune** (b), two stage pretraining + finetune reverts the geometry to a baseline-like state (a), confirming that contrastive feature alignment is destroyed by the regression gradient. Under **CL Joint Training** (c), visual map features and RTK SD map features collapse into a single compact cluster while GNSS SD map features are clearly separated from both—exactly same as our target. This suggests that contrastive feature alignment in the whole training stage is important for visual re-localization improvement in Table 8.

5.6 Reliability Prediction Module

We evaluate the reliability prediction on VectorReLoc dataset by sweeping the predicted reliability threshold $\tau \in \{0.1, \ldots, 0.9\}$ and retaining only samples that exceed it.

Table 9: Effect of the reliability prediction module. Each row corresponds to retaining only samples whose reliability score exceeds a threshold τ. The first row gives an unfiltered baseline.

Threshold τ	Count	Coverage (%)	Position Recall (%) ↑				Orientation Recall (%) ↑				Error ↓			
			1 m	2 m	5 m	10 m	1°	2°	5°	10°	E_{xy} (m)	E_x (m)	E_y (m)	E_{yaw} (°)
Baseline (no filter)	23,278	100.0	30.39	54.62	84.85	97.62	94.62	97.55	99.08	99.75	2.6515	2.3922	0.6792	0.3620
> 0.1	23,271	100.0	30.40	54.63	84.86	97.63	94.64	97.57	99.10	99.76	2.6504	2.3915	0.6781	0.3602
> 0.2	23,264	99.9	30.41	54.65	84.87	97.64	94.67	97.59	99.10	99.75	2.6490	2.3908	0.6771	0.3591
> 0.3	23,249	99.9	30.43	54.68	84.90	97.64	94.71	97.63	99.11	99.75	2.6470	2.3897	0.6753	0.3574
> 0.4	23,240	99.8	30.44	54.70	84.90	97.65	94.72	97.64	99.12	99.76	2.6458	2.3892	0.6743	0.3564
> 0.5	23,214	99.7	30.48	54.76	84.95	97.65	94.80	97.70	99.15	99.76	2.6415	2.3867	0.6710	0.3527
> 0.6	23,003	98.8	30.73	55.15	85.32	97.78	95.43	98.26	99.48	99.87	2.6079	2.3693	0.6442	0.3140
> 0.7	22,907	98.4	30.85	55.31	85.51	97.84	95.72	98.51	99.62	99.93	2.5932	2.3608	0.6334	0.2962
> 0.8	22,777	97.8	30.98	55.50	85.68	97.88	96.02	98.74	99.72	99.97	2.5800	2.3541	0.6219	0.2822
> 0.9	22,528	96.8	**31.13**	**55.70**	**85.79**	**97.90**	**96.27**	**98.93**	**99.79**	**99.98**	**2.5706**	**2.3497**	**0.6142**	**0.2712**

As shown in Table 9, localization metrics generally improve as τ increases, confirming that the module discriminates accurate predictions from inaccurate ones. At τ=0.9, E_{xy} decreases by 3.1% and E_{yaw} by **25.1%** (0.3620° → 0.2712°) relative to the unfiltered baseline, while 1-m position recall and 1 ° orientation recall improve by +0.74 pp and +1.65 pp, respectively. Crucially, even at τ=0.9, **96.8%** of frames (22,528/23,278) remain retained, demonstrating that the model is highly confident on the vast majority of inputs and unreliable predictions are rare.

5.7 Inference Latency and Deployment

High-performance GPUs. On NVIDIA GPUs with identical Argoverse2 inputs (7 cams $\times 768 \times 1024$, batch $= 1$, Table 10), VectorReLoc is both the lightest (**45.42 M** params, **1197.71 G** FLOPs; ∼17%/27% fewer than HOLO [27]) and the fastest, e.g. **65.45 ms** on A100 (1.48× HOLO).

Table 10: Inference latency / speed on Argoverse2 dataset.

Model	**Params** (M)	**FLOPs** (G)	**L20 GPU** (ms / FPS)	**A100 GPU** (ms / FPS)	**A800 GPU** (ms / FPS)
HOLO [27]	54.64	1635.32	132.81 / 7.5	96.80 / 10.3	95.26 / 10.5
MapLocNet [26]	53.99	1397.16	106.28 / 9.4	66.98 / 14.9	65.78 / 15.2
Ours	**45.42**	**1197.71**	**92.44 / 10.8**	**65.45 / 15.3**	**65.35 / 15.3**

Edge computing chips. On the Horizon Robotics J6M chip (Table 11), the re-localization head costs only **5 ms** (Int16) or **2 ms** (Int8), atop a shared ≈44 ms online mapping module reused from perception. Int16 is lossless (†); Int8 drops 1-m recall by 8.9 pp yet keeps half-lane (Δx<1.75 m) recall within 4.7 pp and yaw error below 0.50°, confirming real-time viability.

Fig. 4 shows representative qualitative re-localization results on the VectorReLoc dataset. The VectorReLoc model performs well across diverse complex scenarios.

Table 11: Deployment evaluation on Horizon Robotics J6M chip.

Model	Latency	2D Position				Δx (Lateral)			Δy (Long.)		Yaw	
	(ms) ↓	E_{pos}(m)↓	<1 m Recall (%) ↑	<2 m Recall (%) ↑	<5 m Recall (%) ↑	E_x(m)↓	<1 m Recall (%) ↑	<1.75 m Recall (%) ↑	E_y(m)↓	<2 m Recall (%) ↑	E_{yaw}(°)↓	< 1° Recall (%) ↑
Float32	–	**2.7230**	**26.25**	**51.81**	84.70	**2.3685**	**37.73**	**55.36**	**0.8479**	**90.69**	**0.3852**	**93.77**
Int16†	**5**	2.7327	26.05	51.57	**84.73**	2.3748	37.58	55.24	0.8564	90.54	0.4039	93.67
Int8	**2**	2.9985	17.35	44.65	83.28	2.5432	32.69	50.64	1.0659	86.85	0.4997	91.42

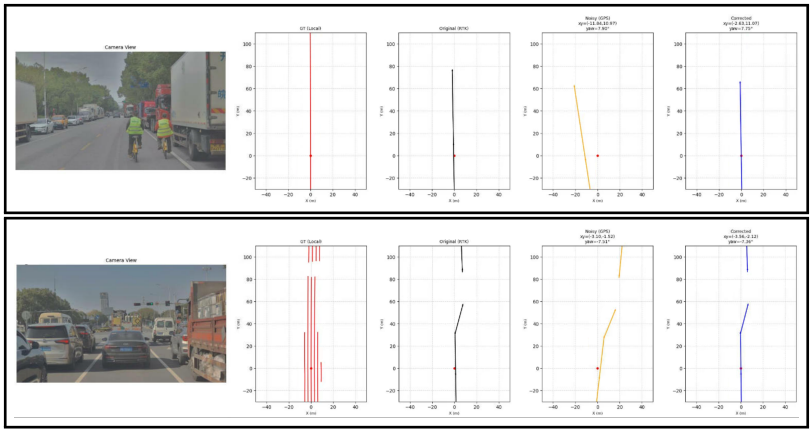


Fig. 4: Qualitative SD map re-localization results on the VectorReLoc dataset. Red: Ground-truth local visual map. Black: Ground-truth RTK SD map. Orange: GNSS-retrieved SD map (before correction). Blue: VectorReLoc-corrected SD map.

6 Conclusion

We introduced VectorReLoc, a sparse vectorized SD map visual re-localization framework with contrastive feature alignment and reliability scoring, avoiding dense BEV raster pipelines. We build a large-scale dataset with paired RTK/GNSS-retrieved SD maps and formulate the GNSS Offset Distribution Hypothesis: real GNSS offsets are heavy-tailed and are well captured by independent Student-t distributions, as validated by cross-evaluation between real and synthetic offsets. Across nuScenes, Argoverse2, and our dataset, VectorReLoc achieves large-margin improvements in recall and error over prior end-to-end methods, and the re-localization head runs in 2–5 ms on edge computing chips. In future work, VectorReLoc can explore richer semantics and be extended to more scenarios of robotics.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 62472033) and the Beijing Natural Science Foundation (Grant No. L2607023). This work was also supported by Bosch High Performance Computing Cluster.

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: CVPR. pp. 11621–11631 (2020)
2. Camiletto, A.B., Bochicchio, A., Liniger, A., Dai, D., Gawel, A.: U-BEV: Height-aware bird's-eye-view segmentation and neural map-based relocalization. In: IROS. pp. 5597–5604. IEEE (2024)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. In: ICML (2020)
4. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: ICML (2016), https://arxiv.org/abs/1506.02142
5. Geifman, Y., El-Yaniv, R.: SelectiveNet: A deep neural network with an integrated reject option. In: ICML (2019), https://arxiv.org/abs/1901.09192
6. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. International Journal of Robotics Research (IJRR) (2013)
7. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: ICML (2017), https://arxiv.org/abs/1706.04599
8. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
9. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: ICLR (2017), https://arxiv.org/abs/1610.02136
10. Kendall, A., Gal, Y.: What uncertainties do we need in Bayesian deep learning for computer vision? In: NeurIPS (2017), https://arxiv.org/abs/1703.04977
11. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning (2020), https://arxiv.org/abs/2004.11362
12. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: NeurIPS (2017), https://arxiv.org/abs/1612.01474
13. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In: NeurIPS (2018), https://arxiv.org/abs/1807.03888
14. Li, T., Jia, P., Wang, B., Chen, L., Jiang, K., Yan, J., Li, H.: LaneSegNet: Map learning with lane segment perception for autonomous driving (2024), https://arxiv.org/abs/2312.16108
15. Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks. In: ICLR (2018), https://arxiv.org/abs/1706.02690
16. Liao, B., Chen, S., Wang, X., Cheng, T., Zhang, Q., Liu, W., Huang, C.: MapTR: Structured modeling and learning for online vectorized HD map construction (2023), https://arxiv.org/abs/2208.14437
17. Liao, B., Chen, S., Zhang, Y., Jiang, B., Zhang, Q., Liu, W., Huang, C., Wang, X.: MapTRv2: An end-to-end framework for online vectorized HD map construction (2024), https://arxiv.org/abs/2308.05736
18. Liu, Z., Zhang, X., Liu, G., Zhao, J., Xu, N.: Leveraging enhanced queries of point sets for vectorized map construction (2024), https://arxiv.org/abs/2402.17430
19. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. ICLR (2019)
20. Luo, K.Z., Weng, X., Wang, Y., Wu, S., Li, J., Weinberger, K.Q., Wang, Y., Pavone, M.: Augmenting lane perception and topology understanding with standard definition navigation maps. In: ICRA. pp. 4029–4035. IEEE (2024)

21. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding (2018), https://arxiv.org/abs/1807.03748
22. OpenStreetMap contributors: OpenStreetMap. https://www.openstreetmap.org (2004), accessed: 2026-02-22
23. Sarlin, P.E., DeTone, D., Yang, T.Y., Avetisyan, A., Straub, J., Malisiewicz, T., Bulo, S.R., Newcombe, R., Kontschieder, P., Balntas, V.: OrienterNet: Visual Localization in 2D Public Maps with Neural Matching. In: CVPR (2023)
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS. vol. 30 (2017)
25. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Kaesemodel Pontes, J., Ramanan, D., Carr, P., Hays, J.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. In: Vanschoren, J., Yeung, S. (eds.) NeurIPS. vol. 1 (2021)
26. Wu, H., Zhang, Z., Lin, S., Mu, X., Zhao, Q., Yang, M., Qin, T.: MapLocNet: Coarse-to-fine feature registration for visual re-localization in navigation maps. In: IROS. pp. 13198–13205 (2024)
27. Zhong, X., Cao, X., Feng, J., Fang, H.: HOLO: Homography-guided pose estimator network for fine-grained visual localization on SD maps (2026), https://arxiv.org/abs/2601.02730
28. Zhou, Z., Qi, Z., Cheng, L., Xiong, G.: SegLocNet: Multimodal localization network for autonomous driving via bird's-eye-view segmentation (2025), https://arxiv.org/abs/2502.20077