SCORE: Saturated Consensus Relocalization in Semantic Line Maps

Haodong Jiang*[1], Xiang Zheng*[1], Yanglin Zhang*[1], Qingcheng Zeng[2], Yiqian Li[1], Ziyang Hong[1], Junfeng Wu[1]

***Abstract*— This is the arXiv version of our paper published on IROS 2025. We present SCORE, a visual relocalization system that achieves unprecedented map compactness by adopting semantically labeled 3D line maps. SCORE requires only 0.01%–0.1% of the storage needed by structure-based or learning-based baselines, while maintaining practical accuracy and comparable runtime. The key innovation is a novel robust estimation mechanism, *Saturated Consensus Maximization* (Sat-CM), which generalizes classical *Consensus Maximization* (CM) by assigning diminishing weights to inlier associations according to maximum likelihood with probabilistic justification. Under extreme outlier ratios (up to 99.5%) arising from one-to-many ambiguity in semantic matching, Sat-CM enables accurate estimation when CM fails. To ensure computational efficiency, we propose an accelerating framework for globally solving Sat-CM formulations and specialize it for the Perspective-n-Lines problem at the core of SCORE.**

I. Introduction

Visual relocalization refers to the task of estimating a camera's pose in a known environment from an input image—a critical capability for mobile robotics, augmented reality, and related applications. This process relies on stored scene representations, which vary across methods: some employ 3D map points with visual descriptors [1]–[5], others leverage deep feature maps from posed reference images [6], [7], or encode scene geometry into weights of neural networks [8], [9]. While these paradigms differ in how they represent and utilize scene information, they face a shared trade-off between representation compactness and estimation accuracy. In this work, we advance towards unprecedented compactness while maintaining practical accuracy with two ingredients as illustrated in Fig. 1: a semantically labeled 3D line map as scene representation, and *Saturated Consensus Maximization* (Sat-CM) as an enabling robust mechanism for resolving one-to-many ambiguous association. We adopt 3D lines to represent scene geometry due to their ubiquity in man-made environments and ability to capture structural elements more compactly than points. And we adopt semantics as the visual descriptor of map lines for three reasons. Firstly, semantics is a naturally compressed descriptor which takes only an integer label to store given the dictionary, consuming far less memory than fine-grained descriptors like SIFT [10] and SuperPoint [11]. Secondly, semantics exhibit strong invariance to partial occlusions and viewpoint changes, which are common challenges to line descriptors [12]. Lastly, The increasing availability of accurate segmentation models (e.g., SAM2 [13]) and vision-language systems (e.g., GPT-4 [14]) makes semantic labeling increasingly practical.

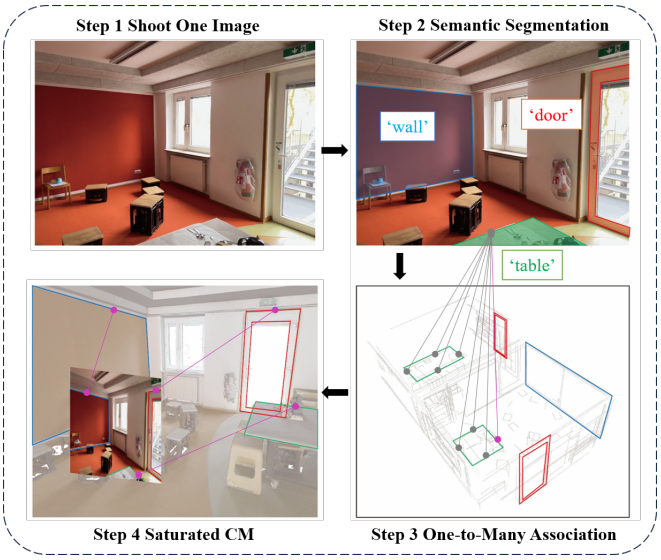


Fig. 1: A kidnapped robot relocalizes itself by associating 2D lines in an image and 3D map lines based on semantic labels, and solving a perspective-n-lines problem. We propose **Saturated Consensus Maximization** to address the extremely high outlier ratio caused by one-to-many associations.

However, this highly compact approach introduces a critical challenge: associating 2D-3D lines via semantic labels causes one-to-many ambiguity and extreme outlier ratios (up to 99.5% in our experiments) for pose estimation. Although prior works have acknowledged similar ambiguity when using simple or quantized visual descriptors [1]–[3], the field still lacks a principled methodology for handling one-to-many association. Current approaches typically avoid rather than address association ambiguity - for instance, the widely-used ratio test [10] explicitly rejects ambiguous matches by requiring a dominant best candidate. In contrast, we embrace ambiguous associations through a principled robust mechanism, Sat-CM, which generalizes the classical *consensus maximization* (CM) method by assigning a diminishing weight for inlier associations according to likelihood. To integrate Sat-CM into our relocalization pipeline, named SCORE, we develop a general accelerated global search

*:Equal contribution. [1]Haodong Jiang, Xiang Zheng, Yanglin Zhang, Yiqian Li, Ziyang Hong, and Junfeng Wu are with School of Data Science, The Chinese University of Hong Kong, Shenzhen, P. R. China, {haodongjiang, 224045013, 119010446,yiqianli}@link.cuhk.edu.cn, {hongziyang,junfengwu}@cuhk.edu.cn. [2] Qingcheng Zeng is with Robotics and Autonomous Systems Thrust, System Hub, The Hong Kong University of Science and Technology(Guangzhou), P. R. China, qzeng450@connect.hkust-gz.edu.cn.

framework for Sat-CM problems, and devise a specialized solution for the perspective-n-lines (PnL) problem. We summarize our **key contributions** as follows:

1) **Ultra-Compact Visual Relocalization:** we push the boundaries of map compactness for visual relocalization by adopting a semantically labeled 3D line map.
2) **Novel Robust Mechanism:** to address one-to-many ambiguity inherent in semantic association, we propose the Sat-CM method which generalizes classic CM method and evaluates inlier associations according to the maximum likelihood criteria.
3) **Accelerated Global Solver:** we develop an accelerated global search framework for general Sat-CM problems, and apply it to solve the PnL problem central to our pipeline based on rigorous interval analysis.
4) **Comprehensive Evaluation:** we demonstrate superiority of Sat-CM over CM, and evaluate practicality of SCORE with extensive experiments on the ScanNet++ [15] dataset. Enabled with the power of Sat-CM and an accelerated global search algorithm, SCORE achieves practical accuracy within comparable runtime, while consuming only 0.01% to 0.1% storage of representative baselines.

II. Related Work

A. Visual Relocalization and Storage Burden

Structure-based visual relocalization methods establish explicit 2D-3D associations through detected local point features [10], [11] and their matches [16], often aided by image retrieval (IR) techniques [17]. Recent advances [18]–[20] incorporate line segments to exploit structural regularities in man-made environments. While accurate, these methods maintain memory-intensive 3D maps with heavy visual descriptors (128 bytes per SIFT or 1 KB per SuperPoint descriptor vs. 12 bytes per 3D coordinate). **End-to-end learning methods** bypass explicit association by encoding scenes in neural network weights [8], [9]. Though elegant, these approaches are inherently scene-specific, requiring adaptation for new environments. Hybrid solutions [6], [7] mitigate this issue by regressing poses relative to retrieved reference images with pre-established 2D-3D associations. However, they still demand substantial storage for either the reference images or their deep feature maps, which consume hundreds of KB per image even after quantization [7].

B. Compact Representation and One-to-Many Ambiguity

To relieve the burden of storing memory-intensive visual descriptors, previous works propose to keep an informative subset of the 3D quantities [21], quantize the visual descriptors [1], [2], or adopt a hybrid approach [3], [4] to reduce the map size. The one-to-many ambiguity in association arises along with descriptor quantization, while existing solutions are fundamentally limited. Sattler et al. [1] prune non-unique associations using co-visibility graphs, but risk prematurely discarding real associations. Recent RANSAC variants [2], [3] generate hypotheses exclusively from sampling unambiguous correspondences and utilize ambiguous associations only for evaluating the pose hypotheses. These variants fail when unique associations are unavailable and remain constrained by a CM formulation, which is not suitable under one-to-many ambiguity.

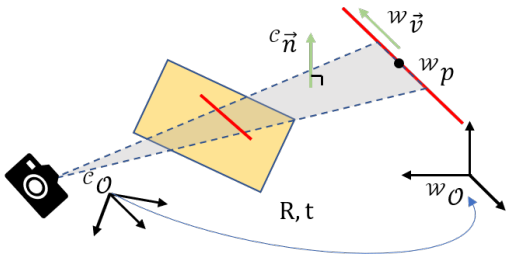


Fig. 2: Projection model underlying the PnL problem.

III. Preliminary

Notations: we use the notation $\mathbf{a} \bullet \mathbf{b} := \mathbf{a}^\top \mathbf{b}$, and use $(\mathbf{a}, \mathbf{b}) := \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}$ to denote the concatenation of two vectors. We use the notation ${}^{\mathcal{F}}\mathbf{a}$ to highlight that $\mathbf{a}$ is observed in the reference frame $\{\mathcal{F}\}$. Specifically, we denote the normalized camera frame as $\mathcal{C}$, and the world frame as $\mathcal{W}$. We use $\mathbf{1}\{\cdot\}$ to denote the indicator function, which equals one if the event inside the bracket happens and equals zero otherwise.

Consider a 2D line in the image which writes as follows in the pixel coordinate: $\begin{bmatrix} A & B & C \end{bmatrix}(u, v, 1) = 0$. Given the camera intrinsic matrix $\mathbf{K}$, we can write it in the normalized image coordinate as $\begin{bmatrix} A_c & B_c & C_c \end{bmatrix}(x, y, 1) = 0$, with $\begin{bmatrix} A_c & B_c & C_c \end{bmatrix} = \begin{bmatrix} A & B & C \end{bmatrix}\mathbf{K}$. We use the normalized coefficient vector ${}^{\mathcal{C}}\vec{\mathbf{n}}$ to parameterize a 2D line l in $\{\mathcal{C}\}$:

$$^{\mathcal{C}}\vec{\mathbf{n}} = \frac{(A_c, B_c, C_c)}{\|(A_c, B_c, C_c)\|}, \quad l := \{^{\mathcal{C}}\mathbf{p} \in \mathbb{R}^2 | {}^{\mathcal{C}}\vec{\mathbf{n}} \bullet ({}^{\mathcal{C}}\mathbf{p}, 1) = 0\}.$$

We refer to ${}^{\mathcal{C}}\vec{\mathbf{n}}$ as the normal vector since it is perpendicular to the plane passing through the camera origin and l. As for a 3D line L observed in the world coordinate, we parameterize it with a point ${}^{\mathcal{W}}\mathbf{p}_0 \in \mathbb{R}^3$ and a direction vector ${}^{\mathcal{W}}\vec{\mathbf{v}} \in \mathbb{S}^2$:

$$L := \{^{\mathcal{W}}\mathbf{p} \in \mathbb{R}^3 | ({}^{\mathcal{W}}\mathbf{p} - {}^{\mathcal{W}}\mathbf{p}_0) \times {}^{\mathcal{W}}\vec{\mathbf{v}} = 0\}.$$

Assume the relative transformation from the normalized camera frame to the world coordinate writes as follows

$$^{\mathcal{W}}_{\mathcal{C}}\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}.$$

Assume a 2D line l in the image is the projection of a 3D line L in the world map, the following two equations [22] uniquely determine the projection:

$$(\mathbf{R}\,{}^{\mathcal{C}}\vec{\mathbf{n}}) \bullet {}^{\mathcal{W}}\vec{\mathbf{v}} = 0, \tag{1}$$

$$(\mathbf{R}\,{}^{\mathcal{C}}\vec{\mathbf{n}}) \bullet ({}^{\mathcal{W}}\mathbf{p}_0 - \mathbf{t}) = 0. \tag{2}$$

IV. Sat-CM Exemplified by the Robust PnL Problem with Semantic Matching

We address a challenging scenario where a kidnapped robot relocalizes itself with one image and an aggressively compressed scene map, which consists of 3D lines parameterized by their endpoints and semantic labels. The robot

extracts 2D line segments from the image with semantic labels, and associates with map lines based on semantics. Without any other information, the robot associates each 2D line with all map lines of the same label, leading to a significant outlier ratio. While the CM method [23] traditionally underpins robust estimation in robotics, its power is restricted under such one-to-many ambiguity. We therefore propose the Sat-CM method, which introduces a saturation function to handle more than one inlier associations. In this section, we formalize Sat-CM and motivate it with the PnL problem, where the robot pose is estimated using matched 2D image lines and 3D map lines.

A. Saturated Consensus Maximization

Suppose we are given a sample set $\{\mathbf{x}_k\}$, a data set $\{\mathbf{y}_m\}$, and a residual function $f(\boldsymbol{\Theta}|\mathbf{x},\mathbf{y})$ w.r.t an unknown parameter $\boldsymbol{\Theta}$. Suppose under the ground truth $\boldsymbol{\Theta}^o$, the residual of a real sample-data association satisfies $|f(\boldsymbol{\Theta}^o|\mathbf{x},\mathbf{y})| \leq \epsilon$, where ϵ is a tolerance term to model against the effect of random noise. For each sample $\mathbf{x}_k$, there exists multiple matched data denoted as $\{\mathbf{y}_{m_k^{(i)}}\}$. The Sat-CM method estimates $\boldsymbol{\Theta}^o$ by solving the following problem:

$$\max_{\boldsymbol{\Theta}} \quad \sum_{k=1}^{K} \sigma_k \left(\sum_i \mathbf{1}\{|f(\boldsymbol{\Theta}|\mathbf{x}_k, \mathbf{y}_{m_k^{(i)}})| \leq \epsilon\} \right), \quad (3)$$

where $\sigma_k(N) = \sum_{n=1}^{N} w_k(n)$ is referred to as the *saturation function*, and one arrives at the classic CM formulation with $\sigma_k(N) = N,\ w_k(n) = 1\ \forall n \geq 1$. For conciseness, we introduce several terminologies. Under a parameter hypothesis $\boldsymbol{\Theta}$, we refer to a sample-data pair $(\mathbf{x}_k, \mathbf{y}_{m_k^{(i)}})$ as an *inlier association* given that $|f(\boldsymbol{\Theta}|\mathbf{x}_k, \mathbf{y}_{m_k^{(i)}})| \leq \epsilon$. We use the terminology *settled* to describe the state of a sample $\mathbf{x}_k$ having at least one inlier association, and use *unsettled* to describe the state of $\mathbf{x}_k$ having no inlier association. As the naming indicates, the saturation function assigns a diminishing weight $w_k(n)$ to each additional inlier of $\mathbf{x}_k$, and the total assigned weight saturates to a finite value.

B. PnL Problem under One-to-many Association

In the PnL problem, we treat 2D lines l_k extracted from the image as samples $\mathbf{x}_k$, and the 3D map lines L_m as data $\mathbf{y}_m$. We denote a 2D line with semantic label s_k, and a 3D line with semantic label s_m respectively as:

$$l_k : ({}^{\mathcal{C}}\vec{\mathbf{n}}_k, s_k), \quad L_m : ({}^{\mathcal{W}}\mathbf{p}_m, {}^{\mathcal{W}}\mathbf{v}_m, s_m),$$

A semantic label s is an integer id corresponding to a word in the dictionary, e.g., *(1,'chair')* and *(2,'table')*. For simplicity of notation and implementation, we treat a line with multiple labels as separate lines sharing the same geometric position but with different labels. We match each image line l_k with all map lines of the same label, and denote this set of associations as $\{L_{m_k^{(i)}}\}$, with $\#\{L_{m_k^{(i)}}\} := M_k$.

The common practice in the robust PnL problem adopts the CM method and estimates rotation based on constraint (1) first [24]. Under our setting, this problem writes:

$$\max_{\mathbf{R}\in SO(3)} \quad \sum_{k=1}^{K} \sum_{i=1}^{M_k} \mathbf{1}\{|(\mathbf{R}^{\mathcal{C}}\vec{\mathbf{n}}_k) \bullet {}^{\mathcal{W}}\vec{\mathbf{v}}_{m_k^{(i)}}| \leq \epsilon_r\}, \quad (4)$$

where the outer summation is over all 2D lines l_k, and the inner summation is over the associated 3D lines $L_{m_k^{(i)}}$. As illustrated by the following toy example, CM may flavor an unreasonable estimate under one-to-many ambiguity. Suppose the robot captures an image with 10 lines detected, and two rotation hypotheses $\mathbf{R}_1$ and $\mathbf{R}_2$ are evaluated by (4). Under hypothesis $\mathbf{R}_1$, lines l_1 to l_9 each has one inlier while line l_{10} has no inlier, resulting a value of 9. Under hypothesis $\mathbf{R}_2$, lines l_1 to l_9 have no inlier, while line l_{10} has 10 inliers, yielding a value of 10. Although $\mathbf{R}_2$ achieves a higher value in (4), one would intuitively consider $\mathbf{R}_1$ to be the better hypothesis since more 2D lines are settled under $\mathbf{R}_1$, indicating that a more diverse set of constraints—arising from the projections of these lines—are possibly satisfied. In case of $\mathbf{R}_2$, while l_{10} yields 10 inliers, at most one of them is valid, and the inliers from fake associations do not contribute valid constraints on the rotation.

In pursuit of a systematic solution to the above issue, we adopt Sat-CM (3) and formulate the following problem:

$$\max_{\mathbf{R}\in SO(3)} \quad \sum_{k=1}^{K} \sigma_k \left(\sum_{i=1}^{M_k} \mathbf{1}\{|(\mathbf{R}^{\mathcal{C}}\vec{\mathbf{n}}_k) \bullet {}^{\mathcal{W}}\vec{\mathbf{v}}_{m_k^{(i)}}| \leq \epsilon_r\} \right). \quad (5)$$

The intuition from the previous toy example—that settling more 2D lines is better—may lead one to adopt the following 'truncated' saturation function, which essentially counts the number of settled 2D lines: let $\sigma_k(N) = \sum_{n=1}^{N} w_k(n)$,

$$\sigma_k(N) = \mathbf{1}\{N \geq 1\},\ w_k(n) = 1\{n = 1\}. \quad (6)$$

However, it performs poorly in highly ambiguous scenarios where bad hypotheses may tie (or even beat) the good ones in the number of settled 2D lines. This occurs because the intuition underlying (6) fails to account for other important factors such as the number of associations M_k for each 2D line l_k. For example, settling 2 lines each with 1 inlier out of 1000 putative associations is less likely to contribute to an accurate orientation than settling a single line with 1 inlier out of 2 putative associations. This underscores the necessity of principled saturation function design methodology.

C. Likelihood-Based Saturation Function Design

From a maximum likelihood standpoint, we propose a saturation function as follows: let $\sigma_k(N) = \sum_{n=1}^{N} w_k(n)$:

$$\sigma_k(N) = \log(1 + C\frac{N}{M_k}),\ w_k(n) = \log(\frac{M_k + nC}{M_k + (n-1)C}), \quad (7)$$

where C is a scaling constant to be derived in likelihood later. For each 2D line l_k, we introduce a hidden variable ι_k to indicate index of the real association: $\iota_k = i$ for association $(l_k, L_{m_k^{(i)}})$ to be real, while $\iota_k = 0$ for all associations to be fake. Denote residual resulting from a hypothesis $\mathbf{R}$ and association $(l_k, L_{m_k^{(i)}})$ as $r_{k,i}(\mathbf{R}) := |(\mathbf{R}^{\mathcal{C}}\vec{\mathbf{n}}_k) \bullet {}^{\mathcal{W}}\vec{\mathbf{v}}_{m_k^{(i)}}|$. Our derivations are based on two assumptions:

Assumption 1: For each 2D line l_k, the association set $\{(l_k, L_{m_k^{(i)}})\}$ contains the real association with probability $q > 0$, and each association is equally likely to be real, i.e.,

$$\begin{cases} Pr(\iota_k = 0) = 1 - q, \\ Pr(\iota_k = i) = q/M_k, \quad 1 \le i \le M_k. \end{cases}$$

Assumption 2: With the ground truth rotation $\mathbf{R}^o$,

$$\begin{cases} r_{k,i}(\mathbf{R}^o) \sim U(0, \epsilon_r), & \text{conditioned on } \iota_k = i \\ r_{k,i}(\mathbf{R}^o) \sim U(0, u_r), & \text{conditioned on } \iota_k \neq i. \end{cases}$$

where ϵ_r is the same tolerance used in (5), u_r is the residual upper bound, and $u_r = 1$ in the PnL rotation problem.

Note that Assumption 1 applies when the associations have equal confidence level, and Assumption 2 extends the probabilistic justification of CM [25]. We calculate the likelihood for line l_k under a hypothesis $\mathbf{R}$ by marginalizing the hidden variable ι_k based on the above two assumptions. Denote the index set of the N_k inliers under $\mathbf{R}$ as $\mathcal{N}_k$, with $N_k = \#\mathcal{N}_k$, and the vector stacking all residuals $r_{k,i}(\mathbf{R})$'s as $\mathbf{r}_k$, where $i = 1, \cdots, M_k$. Then, the likelihood for line l_k writes

$$\begin{aligned} l_{\mathbf{R}}(\mathbf{r}_k) =& p(\mathbf{r_k}, \iota_k = 0) + \sum_{\iota_k \in \mathcal{N}_k} p(\mathbf{r_k}, \iota_k) + \sum_{\iota_k \notin \mathcal{N}_k} p(\mathbf{r_k}, \iota_k), \\ =& (1-q) u_r^{-M_k} + N_k q / M_k u_r^{-(M_k-1)} \epsilon_r^{-1} + 0 \\ =& u_r^{-M_k} (1-q)(1 + C N_k / M_k), \end{aligned}$$

where $C := u_r \epsilon_r^{-1} q (1-q)^{-1}$. We arrive at (7) by taking logarithm and subtracting a common constant (given u_r=1) in the likelihood of different l_k: $\log(1-q) - M_k \log(u_r)$.

We highlight two key observations on (7). **First,** the weight $w_k(n)$ assigned to the nth inlier association decreases w.r.t the total association number M_k, essentially penalizing the level of ambiguity. **Second,** (7) assigns a diminishing weight $w_k(n)$ to inliers, and the total assignment saturates to $\log(1 + C)$ despite of M_k. This mechanism ensures that additional inliers contribute less to the overall consensus score, and prioritizes settling more lines over inflating the number of insignificant inliers. Although the diminishing rate of $w_k(n)$ is controlled by a hyperparameter q, estimators based on (7) are not sensitive to the choice of q in an appropriate range, as supported by our relocalization experiments. To enclose our discussion on function design, we present the global landscapes of the CM rotation problem (4) and the Sat-CM one (5) in Fig. 3, where the 2D/3D associations come from one of the query image in our experiments. As indicated by Fig. 3, adopting Sat-CM with a likelihood-justified function design structurally reinforce the true value as the global optimum while suppressing ambiguous solutions.

Finally, given the global optimum $\hat{\mathbf{R}}_\sigma^*$ of (5) and the corresponding inliers, we further formulate the translation estimation problem using Sat-CM:

$$\max_{\mathbf{t} \in \mathbb{R}^3} \sum_{k=1}^{K} \sigma_k \left(\sum_{i=1}^{M_k} \mathbf{1} \left\{ |{}^{\mathcal{W}}\vec{\mathbf{n}}_k^* \bullet ({}^{\mathcal{W}}\mathbf{p}_{m_k^{(i)}} - \mathbf{t})| \le \epsilon_t \right\} \right), \quad (8)$$

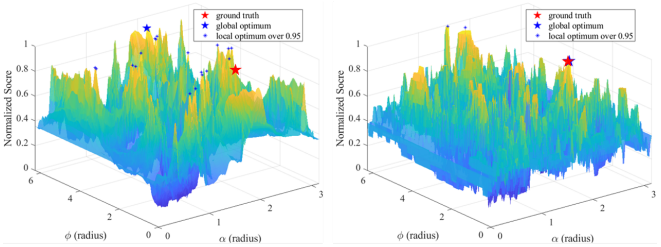


Fig. 3: The normalized global landscapes under (a) the CM problem (4) and (b) the Sat-CM problem (5). Each point is evaluated with a fixed rotation axis under polar coordinates (ϕ, α) and the ad-hoc optimal rotation amplitude. We choose (7) as the saturation function.

where ${}^{\mathcal{W}}\vec{\mathbf{n}}_k^* := \hat{\mathbf{R}}_\sigma^{*\mathcal{C}} \vec{\mathbf{n}}_k$ is the rotated normal vector[1].

V. Solving Sat-CM Efficiently with Dimension-Reduced Branch-and-Bound

As an extension of the NP-hard CM problem [26], Sat-CM is also computationally challenging to solve. In this section, we propose an accelerated branch-and-bound (BnB) algorithm for Sat-CM problems, which reduces the branching dimensions by one through novel bounding techniques as inspired by a recent accelerating framework [27]. Next, we deploy this general algorithm to solver the PnL problems (5) and (8). While a recent work [28] proposes a similar global rotation solver for PnL, our FGO-PnL (Fast and Globally Optimal) solver provides three key advances. **Firstly**, FGO-PnL accommodates the more general Sat-CM formulation. **Secondly**, rigorous interval analysis stands behind FGO-PnL to guarantee bound validity, while [28] uses heuristic approximation in their implementation. **Lastly,** FGO-PnL also includes an accelerated global translation solver. We release in our Git-Hub repository[2] Matlab and parallelized C++ implementations for FGO-PnL, which provide modular saturation function interfaces for easy customization and backward compatibility with CM via choosing $\sigma(N) = N$.

A. Accelerated Branch-and-Bound for Sat-CM

For clarity, we use $\mathbf{a}^{(k,m_k^{(i)})} := (\mathbf{x}_k, \mathbf{y}_{m_k^{(i)}})$ to denote an association. We start with a 1D Sat-CM problem:

$$\max_{\theta} \sum_{k=1}^{K} \sigma_k \left(\sum_{i=1}^{M_k} \mathbf{1}\{|f(\theta|\mathbf{a}^{(k,m_k^{(i)})})| \le \epsilon\} \right). \quad (9)$$

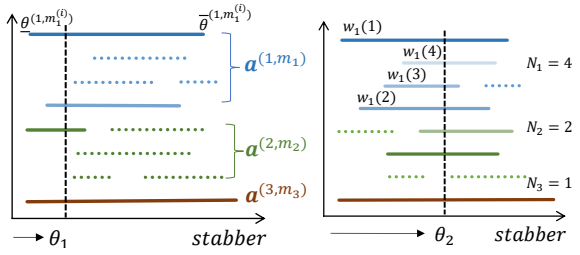
Assume the residual function is continuous in the parameter θ, we obtain by interval analysis:

$$|f(\theta|\mathbf{a}^{(k,m_k^{(i)})})| \le \epsilon \Leftrightarrow \theta \in [\underline{\theta}^{(k,m_k^{(i)})}, \bar{\theta}^{(k,m_k^{(i)})}],$$

in which we assume a single interval is obtained for notational clarity. In practice, an association $\mathbf{a}^{(k,m_k^{(i)})}$ may

[1]Note that since ${}^{\mathcal{W}}\vec{\mathbf{n}}_k$ is not perfectly orthogonal to the direction vector ${}^{\mathcal{W}}\vec{\mathbf{v}}_{k_j}$, choosing different point ${}^{\mathcal{W}}\mathbf{p}_{k_j}$ on a map line influences the residual. In order to make problem (8) invariant with the choice of parameterizing point, we project ${}^{\mathcal{W}}\vec{\mathbf{n}}_i^*$ onto the null space of ${}^{\mathcal{W}}\vec{\mathbf{v}}_{k_j}$ in implementation.

[2]https://github.com/LIAS-CUHKSZ/SCORE



(a) (b)

Fig. 4: Saturated interval stabbing. The increased transparency of intervals represents adaptively decreased weights for each additional inlier. Algorithm 1 sweeps the range of stabber θ from left to right and record the highest value achieved along the way. (a): $v(\theta_1) = \sigma_1(2) + \sigma_2(1) + \sigma_3(1)$, (b):$v(\theta_2) = \sigma_1(4) + \sigma_2(2) + \sigma_3(1)$.

induce multiple disjoint intervals, while it does not affect the proposed algorithm. Based on these intervals, we arrive at an equivalent problem for (9), which we refer to as saturated interval stabbing (Sat-IS) and illustrate in Fig. 4:

$$\max_{\theta} \sum_{k=1}^{K} \sigma_k \left(\sum_{i=1}^{M_k} \mathbf{1}\{\theta \in [\underline{\theta}^{(k,m_k^{(i)})}, \bar{\theta}^{(k,m_k^{(i)})}]\} \right). \tag{10}$$

Denote $M := \sum M_k$, we can solve (10) with $O(M)$ space and $O(M \log M)$ time using Algorithm 1.

Algorithm 1 Saturated Interval Stabbing

Inputs: Intervals $\mathcal{I}_k = \{[\underline{\theta}^{(k,m_k^{(i)})}, \bar{\theta}^{(k,m_k^{(i)})}]\}_{m_k^{(i)}}$ for $k = 1, ..., K$ and functions $\sigma_k(N) = \sum_{n=1}^{N} w_k(n)$.
Outputs: optimal stabbers $\theta^*_{\{\ \}}$, optimal value v^*.

I = sort endpoints in $\{\mathcal{I}_k\}_{k=1}^{K}$.
$v^* = 0,\ v = 0,\ N_k = 0\ (k = 1, ..., K)$.
for $i = 1$ to $len(I)$ **do**
  if $I(i)$ is a left endpoint from $\mathcal{I}_k$ **then**
    $N_k = N_k + 1, \quad v = v + w_k(N_k)$.
    if $c > v^*$ **then**
      $v^* = v, \quad \theta^*_{\{\ \}} = [I(i), I(i+1)]$.
    end if
  else
    $v = v - w_k(N_k), \quad N_k = N_k - 1$.
  end if
end for
return $\theta^*_{\{\ \}}, v^*$

As for a high-dimensional problem (3), we distinguish one parameter θ with others: $\mathbf{\Theta} = (\mathbf{\Theta}_{:-1}, \theta)$, and branches only over the space of $\mathbf{\Theta}_{:-1}$, as motivated by [27]. Denote the optimal value of (3) for $\mathbf{\Theta}_{:-1}$ constricted in a sub-cube $\mathcal{C}$ and θ free as $r^*(\mathcal{C})$. A **lower bound** for $r^*(\mathcal{C})$ writes:

$$\max_{\theta} \sum_{k=1}^{K} \sigma_k \left(\sum_{i=1}^{M_k} \mathbf{1}\{|f(\theta|\mathbf{\Theta}^{(c)}_{:-1}, \mathbf{a}^{(k,m_k^{(i)})})| \le \epsilon\} \right), \tag{11}$$

where $\mathbf{\Theta}^{(c)}_{:-1}$ is the center point of sub-cube $\mathcal{C}$. Notice that the lower bound (11) corresponds to a 1-D Sat-CM problem, and it can be efficiently solved by Algorithm 1. As for the upper bound, we manage to find two bounding functions for each residual term:

$$f_L^{(k,i)}(\theta) \le f(\theta|\mathbf{\Theta}_{:-1}, \mathbf{a}^{(k,m_k^{(i)})}) \le f_U^{(k,i)}(\theta),$$

where the inequality holds for any $\mathbf{\Theta}_{:-1} \in \mathcal{C}$. Given f_L and f_U, we find an **upper bound** for $r^*(\mathcal{C})$ as follows:

$$\max_{\theta} \sum_{k=1}^{K} \sigma_k \Big(\sum_{i=1}^{M_k} \mathbf{1}\{f_L^{(k,i)}(\theta) \le \epsilon \ \textbf{and}\ f_U^{(k,i)}(\theta) \ge -\epsilon\} \Big). \tag{12}$$

Similar to the lower bound, we obtain the upper bound (12) by solving a Sat-IS problem with Algorithm 1. Equipped with the above lower and upper bounds, we search the global optimum of a N-D Sat-CM problem by branching and bounding only a subspace of $N - 1$ dimensions. For details in the BnB procedure, one can refer to our implementation and [27]. Next, we focus on finding the upper bounding functions f_L and f_U for the PnL problem. We put analysis corresponding to the translation part in Appendix IV, and focus on the challenging rotation problem. For conciseness, in the rest of this section, we refer to a general association as $\mathbf{a} := (l, L)$, and denote the corresponding normal, direction vector and a 3D point on L respectively as $\vec{\mathbf{n}}_a$, $\vec{\mathbf{v}}_a$ and $\mathbf{p}_a$.

B. Find Bounding Functions for Rotation Problem in PnL

We parameterize rotation with a rotation axis $\vec{\mathbf{u}} \in \mathbb{S}^2$ and an amplitude $\theta \in [0, \pi]$. We choose θ as the distinguished parameter, and further parameterize $\vec{\mathbf{u}}$ by polar coordinates:

$$\vec{\mathbf{u}} = (\sin\alpha\cos\phi, \sin\alpha\sin\phi, \cos\alpha) \quad \alpha \in [0, \pi]\ \ \phi \in [0, 2\pi].$$

We denote a sub-cube for axis $\vec{\mathbf{u}}$ in the polar coordinate as

$$\mathcal{C}_{\vec{\mathbf{u}}} := \{(\alpha, \phi) | \alpha \in [\underline{\alpha}, \bar{\alpha}], \phi \in [\underline{\phi}, \bar{\phi}]\},$$

and denote its boundary as $\partial\mathcal{C}_{\vec{\mathbf{u}}}$.

Rewrite the residual function $f(\theta, \vec{\mathbf{u}}|\mathbf{a})$ for (5) as

$$\vec{\mathbf{n}}_a^\top \vec{\mathbf{v}}_a + \sin\theta \vec{\mathbf{n}}_a^\top (\vec{\mathbf{u}} \times \vec{\mathbf{v}}_a) + (1 - \cos\theta)\vec{\mathbf{n}}_a^\top [\vec{\mathbf{u}}]_\times^2 \vec{\mathbf{v}}_a, \tag{13}$$

where $[\vec{\mathbf{u}}]_\times$ denotes the skew matrix of $\vec{\mathbf{u}}$. Notice that as a function of $\theta \in [0, \pi]$, the residual (13) is monotone w.r.t:

$$h_1(\vec{\mathbf{u}}|\mathbf{a}) := \vec{\mathbf{u}}^\top (\vec{\mathbf{v}}_a \times \vec{\mathbf{n}}_a), \qquad h_2(\vec{\mathbf{u}}|\mathbf{a}) := \vec{\mathbf{n}}_a^\top [\vec{\mathbf{u}}]_\times^2 \vec{\mathbf{v}}_a.$$

Based on monotonicity, we obtain the bounding functions for $f(\theta, \vec{\mathbf{u}}|\mathbf{a}), \vec{\mathbf{u}} \in \mathcal{C}$ by finding the lower and upper values of spherical functions h_1 and h_2 for $\vec{\mathbf{u}} \in \mathcal{C}_{\vec{\mathbf{u}}}$, denoted as h_1^L, h_1^U, h_2^L and h_2^U. The bounding functions write:

$$f_L(\theta) = \vec{\mathbf{n}}_a^\top \vec{\mathbf{v}}_a + h_1^L \sin\theta + h_2^L (1 - \cos\theta), \tag{14a}$$

$$f_U(\theta) = \vec{\mathbf{n}}_a^\top \vec{\mathbf{v}}_a + h_1^U \sin\theta + h_2^U (1 - \cos\theta). \tag{14b}$$

We summarize our results in the following two theorems. For conciseness, we first introduce several notations:

$$\vec{\mathbf{m}}_a = \frac{\vec{\mathbf{v}}_a + \vec{\mathbf{n}}_a}{\|\vec{\mathbf{v}}_a + \vec{\mathbf{n}}_a\|}, \quad \vec{\mathbf{m}}_a^\perp = \frac{\vec{\mathbf{v}}_a - \vec{\mathbf{n}}_a}{\|\vec{\mathbf{v}}_a - \vec{\mathbf{n}}_a\|}, \quad \vec{\mathbf{c}}_a := \frac{\vec{\mathbf{v}}_a \times \vec{\mathbf{n}}_a}{\|\vec{\mathbf{v}}_a \times \vec{\mathbf{n}}_a\|}.$$

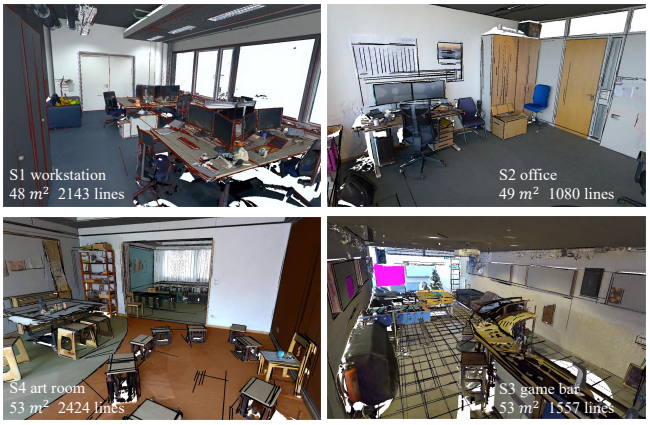


Fig. 5: Meshes of the selected four scenes from ScanNet++ and overlapped with the constructed line map.

Theorem 1 (Extreme Points for $h_2(\vec{\mathbf{u}}|\mathbf{a})$):

1) If $\pm\,\vec{\mathbf{m}}_a \in \boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}$, $\arg\max_{\vec{\mathbf{u}}\in\boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}} h_2(\vec{\mathbf{u}}|\mathbf{a}) = \pm\vec{\mathbf{m}}_a$.
2) If $\pm\,\vec{\mathbf{m}}_a^{\perp} \in \boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}$, $\arg\min_{\vec{\mathbf{u}}\in\boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}} h_2(\vec{\mathbf{u}}|\mathbf{a}) = \pm\vec{\mathbf{m}}_a^{\perp}$.
3) Otherwise, the extreme points fall on $\partial\boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}$.

Theorem 2 (Extreme Points for $h_1(\vec{\mathbf{u}}|\mathbf{a})$):

1) If $\vec{\mathbf{c}}_a \in \boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}$, $\arg\max_{\vec{\mathbf{u}}\in\boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}} h_1(\vec{\mathbf{u}}|\mathbf{a}) = \vec{\mathbf{c}}_a$.
2) If $-\vec{\mathbf{c}}_a \in \boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}$, $\arg\min_{\vec{\mathbf{u}}\in\boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}} h_1(\vec{\mathbf{u}}|\mathbf{a}) = -\vec{\mathbf{c}}_a$.
3) Otherwise, the extreme points fall on $\partial\boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}$.

We present the proof for Theorem 1 in Appendix I and omit the proof for Theorem 2 which is similar. For case 3) in Theorem 2, we actually can explicitly write out the extreme points without the need to transverse $\partial\boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}$, and more explanation is provided in Appendix II.

VI. Experiments on ScanNet++ Dataset

We implement relocalization experiments on the ScanNet++ dataset [15] , mainly for the convenience to construct a semantic line map. ScanNet++ provides for each scene an image sequence captured by an iPhone 13 Pro with default camera setting, and the user can render depth and semantic mask for each image from a human-annotated mesh. For each scene, we split the sequence of images into a reference set and a query set at a 7:1 ratio. We propose a pipeline to construct semantic line maps based on the reference images, and relocalize the query images in the constructed maps. Due to page limit, we present relevant content in Appendix III. As shown in Fig. 5, the quality of the constructed line maps is compromised when compared to the ones constructed by mature point-based pipelines like COLMAP [29], while, on the other hand, it is an ideal testbed for effectiveness of the Sat-CM mechanism and robustness of SCORE.

A. Settings for Relocalization Experiments

We test SCORE with both ground-truth and predicted semantic labels for the query images. Our customized segmentation pipeline integrates RAM++ [30] for semantic tagging and Grounded-SAM [31] for object detection: RAM++ tags images within the line map's dictionary, and Grounded-SAM segments objects based on the recognized words. And we use the released weights without fine-tuning for both models. In the line map, non-stationary object labels are excluded from the dictionary, such as *bottle* and *jacket*. While including these labels might improve relocalization accuracy, their transient nature in real-world environments renders such gains unrealistic. Additionally, we merge semantically ambiguous labels such as *shelf* and *bookshelf* to increase segmentation accuracy. These design choices underscore the critical role of dictionary engineering—a topic meriting deeper investigation in balancing accuracy with practical applicability.

We evaluate our work from two perspectives. From the methodology perspective, we evaluate the proposed Sat-CM mechanism with different saturation functions and compare with the classic CM method. From the practicability perspective, we compare SCORE with baselines in terms of runtime, storage, and accuracy. We respectively choose hloc [5] and PixLoc [6] as representatives for the structure-based methods relying on high-dimensional visual descriptors, and the learning-based method based on deep image features. For fair comparison, all methods use the same splitting of reference and query images and the same image retrieval (IR) results from NetVLAD [17], relocalizing query images within the sub-map observed by 12 retrieved images. We use consistent error thresholds for the FGO-PnL solver across settings: $\epsilon_r = 0.015$ and $\epsilon_t = 0.03$. We run all experiments on a PC equipped with an AMD Core 7950x CPU@4.5GHz, RAM@64GB and a GeForce RTX 5070Ti graphic card.

B. Sat-CM v.s. CM

We present here results of rotation estimation, which encounters the most severe ambiguity as the first step of a cascaded estimation procedure. In FGO-PnL, we utilize the IR results to effectively prune the space of rotation axis $\vec{\mathbf{u}} \in \mathbb{S}^2$ according to that of the first retrieved image. Specifically, we divide $\mathbb{S}^2$ into cubes with equal side length and restrict our search within the cube where the retrieved rotation axis resides in. The side length is chosen according to IR accuracy, and we present the results with side length π (binary division) and $\pi/2$ (octal division), corresponding to a vague and moderate belief in the retrieval accuracy respectively. We use 'PR' and 'GT' in the legends to distinguish using predicted or ground truth semantic labels for the query images. As for the saturation functions, we use SCM0 in the legends for the naive truncated function (6), and SCM1 for the likelihood-based function (7). We choose $q = 0.9$ for SCM1-GT and $q = 0.5$ for SCM1-PR. We record the maximal error when FGO-PnL returns multiple global optimum, while in the complete pipeline, subsequent translation estimation helps select from the rotation candidates. For comparison in translation estimation and sensitivity evaluation in parameter q, we refer the readers to Appendix V.

As evidenced by Table II, SCM1 demonstrates consistent superiority across scenes and settings, and the performance lead is particularly pronounced under harsh conditions involving large search spaces and predicted semantic labels. While SCM0, which uses the naive truncation function (6), yields lower accuracy than CM in most configurations,

	S1	S2	S3	S4
# map lines	2143	1080	1557	2424
# map points	164860	108726	155673	185040
wrong label ratio	17.2%	19.6%	19.4%	20.5%
missed label ratio	27.0%	30.4%	30.7%	30.0%

TABLE I: Scene detailed information.

primarily due to susceptibility to multiple global optima. These results substantiate that Sat-CM, when incorporating a justified saturation function, enables accurate estimation under one-to-many ambiguity when CM fails. Scene S3 (game bar) exhibits the poorest results, which we attribute to compromised quality of the line map due to a cluttered layout, less accurate image pose and depth. We further observe that a more confined search space of rotation axis with side length $\pi/2$ enhances accuracy universally, since ambiguous candidates are pruned ahead of estimation. At last, we highlight the estimation difficulty when using predicted semantic labels, which can not be fully conveyed by the extremely high outlier ratio (consistently over 99%). As presented in Table I, beyond erroneous predictions, missed detections also degrade performance by reducing usable 2D lines.

C. SCORE v.s. Classic Relocalization Pipelines

We report the memory consumption for storing map geometry (3D points or lines), cameras (reference image poses and 2D-3D associations) and features (visual descriptors or deep feature maps). We also report the median runtime which consists of feature extraction, feature association and estimation. For hloc [5], we adopt SuperPoint [11] for keypoint extraction and description (stored in float 16), and use Lightglue [16] for keypoint association. For PixLoc [6], we store the reference images and extract feature maps of the 12 retrieved images online. For SCORE, we adopt the truncated saturation function instead of the likelihood-based function in translation estimation. This is because we further utilize two physical constraints to prune the inliers obtained from solving (8): the truly-associated 3D line resides in front of the camera and its projection intersects with the image. We choose the truncated saturation function in order to keep more translation candidates before pruning, and indeed observe a more robust performance (presented in Appendix V). As the above methods all rely on IR, we include the accuracy achieved by NetVLAD [17] for reference.

As presented in Table III, SCORE achieves practical relocalization accuracy using the true semantic labels, with median errors below 8 cm and 1.5°, though performance degrades when using predicted labels, particularly in translation. While PixLoc and hloc achieve finer-grained accuracy due to their dense point-based representations (Table I), SCORE provides radical memory efficiency (0.01%–0.1% storage overhead) while maintaining comparable estimation runtime thanks to an accelerated global solver and a parallelized C++ implementation. Compared with IR results achieved by NetVLAD, SCORE consistently improve over the median rotation error even when using predicted labels[3]. Given true semantic labels, SCORE achieves better translation accuracy than IR except in S4 which has the most complete reference image database. To sum up, SCORE proves ultra-compact (kB-scale) semantic line maps can sustain viable relocalization, while with semantic segmentation remaining the accuracy and runtime bottleneck.

VII. Conclusion and Future Works

We push the boundary of map compactness in visual relocalization through two key innovations: semantically labeled 3D line maps as an ultra-efficient scene representation, and Saturated Consensus Maximization as a foundational solution for one-to-many ambiguous associations. Experimental validation on ScanNet++ confirms Sat-CM's robustness and demonstrates our pipeline's practical viability despite extreme map compression. To transition toward real-world deployment, we identify three critical research directions. Firstly, develop fast and approximate solvers for Sat-CM to further reduce runtime. Secondly, explore low-dimensional descriptors combining semantic-level efficiency with improved accuracy and repeatability. Thirdly, incorporate both point and line features to offer a more resilient and fine-grained solution.

Acknowledgment

We thank the reviewers for their valuable comments, Mingzhe Li for providing support for baseline implementation, Prof. Jianhua Huang and Prof. Li Jiang for discussion.

References

[1] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys, "Hyperpoints and fine vocabularies for large-scale location recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2102–2110.

[2] N.-T. Tran, D.-K. Le Tan, A.-D. Doan, T.-T. Do, T.-A. Bui, M. Tan, and N.-M. Cheung, "On-device scalable image-based localization via prioritized cascade search and fast one-many ransac," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1675–1690, 2018.

[3] F. Camposeco, A. Cohen, M. Pollefeys, and T. Sattler, "Hybrid scene compression for visual localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7653–7662.

[4] L. Yang, R. Shrestha, W. Li, S. Liu, G. Zhang, Z. Cui, and P. Tan, "Scenesqueezer: Learning to compress scene for camera relocalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8259–8268.

[5] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 716–12 725.

[6] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl *et al.*, "Back to the feature: Learning robust camera localization from pixels to pose," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3247–3257.

[7] J. Revaud, Y. Cabon, R. Brégier, J. Lee, and P. Weinzaepfel, "Sacreg: Scene-agnostic coordinate regression for visual localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 688–698.

[3]Notice that the median rotation error reported in Table III is slightly better than those in Table II, since we use translation estimation to assist selection among multiple rotation candidates.

TABLE II: Compare CM with Sat-CM in rotation estimation.

Method	S1(workstation)	S2(office)	S3(game bar)	S4(art room)	S1	S2	S3	S4	S1	S2	S3	S4
	25%/50%/75% error quantiles (°)				Recall at 5° (%)				Median Outlier Ratio (%)			
CM$^{\pi}$-GT	1.3/3.0/146.9	0.7/1.2/8.3	1.2/12.0/161.5	0.8/1.3/111.0	54	75	45	65				
SCM0$^{\pi}$-GT	1.6/3.5/179.3	1.6/2.7/179.7	2.5/8.0/179.9	2.2/4.9/179.7	56	58	41	51	95.7	94.7	94.9	95.7
SCM1$^{\pi}$-GT	**0.9/1.4/3.1**	**0.6/0.9/1.9**	**0.8/1.7/4.4**	**0.6/0.9/1.7**	**81**	**90**	**76**	**91**				
CM$^{\frac{\pi}{2}}$-GT	1.1/2.4/39.8	0.7/1.1/2.6	1.1/2.8/33.0	0.7/1.1/2.7	63	85	53	79				
SCM0$^{\frac{\pi}{2}}$-GT	1.6/2.9/5.4	1.5/2.2/3.3	2.2/4.7/63.8	1.9/3.2/7.1	72	78	51	68	95.7	94.7	94.9	95.7
SCM1$^{\frac{\pi}{2}}$-GT	**0.8/1.3/2.5**	**0.6/0.9/1.5**	**0.8/1.6/3.7**	**0.6/0.9/1.5**	**90**	**93**	**80**	**95**				
CM$^{\pi}$-PR	2.4/92.5/168.3	1.0/3.9/177.0	7.9/138.8/179.1	1.1/22.5/174.3	31	52	24	45				
SCM0$^{\pi}$-PR	4.6/171.5/179.8	3.0/178.7/179.9	4.2/179.5/180.0	6.6/179.4/180.0	27	33	25	22	99.4	99.5	99.3	99.4
SCM1$^{\pi}$-PR	**1.6/2.6/90.5**	**0.7/1.6/10.8**	**1.1/3.7/149.7**	**0.8/1.8/72.0**	**64**	**73**	**56**	**69**				
CM$^{\frac{\pi}{2}}$-PR	2.0/28.9/53.7	0.9/2.0/18.2	2.6/26.7/61.2	1.0/2.8/41.2	39	68	37	55				
SCM0$^{\frac{\pi}{2}}$-PR	2.8/8.7/89.2	2.2/3.6/17.1	2.9/80.3/103.0	3.0/13.2/102.7	40	61	32	36	99.4	99.5	99.3	99.4
SCM1$^{\frac{\pi}{2}}$-PR	**1.4/2.3/5.3**	**0.7/1.4/2.8**	**1.0/2.4/4.6**	**0.7/1.3/3.6**	**74**	**82**	**76**	**80**				

TABLE III: Baseline comparison in terms of runtime, memory consumption and accuracy.

Method	median runtime	S1	S2	S3	S4	S1	S2	S3	S4
	extract + match&est	storage: geometry&cameras+feature				Median Pose Error ($cm/°$)			
NetVLAD [17]	**25+5 ms**	0+7 MB	0+5 MB	0+6 MB	0+11 MB	7.8/4.5	11.1/10.0	8.8/6.6	3.4/4.0
hloc [5]	30+260 ms	40+788 MB	38+611 MB	51+809 MB	61+951 MB	**0.5/0.2**	0.7/0.2	**0.8/0.2**	**0.5/0.1**
PixLoc [6]	1910+30 ms	40+132 MB	38+90 MB	51+120 MB	61+192 MB	**0.5/0.2**	**0.6/0.2**	**0.8/0.2**	**0.5/0.1**
SCORE$^{\pi}$-PR	1525+245 ms					13.3/2.4	7.7/1.5	100.7/2.5	21.2/1.7
SCORE$^{\frac{\pi}{2}}$-PR	1525+100 ms	**158+17 KB**	**92+8 KB**	**107+12 KB**	**206+19 KB**	10.6/2.2	6.8/1.2	32.3/2.3	14.4/1.2
SCORE$^{\pi}$-GT	/+130 ms					5.9/1.4	3.8/0.8	7.1/1.5	5.0/0.9
SCORE$^{\frac{\pi}{2}}$-GT	/+45 ms					5.9/1.3	3.8/0.8	6.9/1.4	4.9/0.8

[8] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac-differentiable ransac for camera localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6684–6692.

[9] B.-T. Bui, H.-H. Bui, D.-T. Tran, and J.-H. Lee, "Representing 3d sparse map points and lines for camera relocalization," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 8400–8407.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[11] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[12] R. Pautrat, J.-T. Lin, V. Larsson, M. R. Oswald, and M. Pollefeys, "Sold2: Self-supervised occlusion-aware line description and detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 368–11 378.

[13] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[14] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[15] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai, "Scannet++: A high-fidelity dataset of 3d indoor scenes," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

[16] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "Lightglue: Local feature matching at light speed," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 627–17 638.

[17] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[18] S. Liu, Y. Yu, R. Pautrat, M. Pollefeys, and V. Larsson, "3d line mapping revisited," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 445–21 455.

[19] R. Pautrat, I. Suárez, Y. Yu, M. Pollefeys, and V. Larsson, "Gluestick: Robust image matching by sticking points and lines together," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 9706–9716.

[20] H. Liu, C. Cao, H. Ye, H. Cui, W. Gao, X. Wang, and S. Shen, "Lightweight structured line map based visual localization," *IEEE Robotics and Automation Letters*, 2024.

[21] M. Mera-Trujillo, B. Smith, and V. Fragoso, "Efficient scene compression for visual-based localization," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 1–10.

[22] Y. Liu, T. S. Huang, and O. D. Faugeras, "Determination of camera location from 2-d to 3-d line and point correspondences," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 1, pp. 28–37, 1990.

[23] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[24] C. Xu, L. Zhang, L. Cheng, and R. Koch, "Pose estimation from line correspondences: A complete analysis and a series of solutions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1209–1222, 2016.

[25] P. Antonante, V. Tzoumas, H. Yang, and L. Carlone, "Outlier-robust estimation: Hardness, minimally tuned algorithms, and applications," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 281–301, 2021.

[26] T.-J. Chin, Z. Cai, and F. Neumann, "Robust fitting in computer vision: Easy or hard?" in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 701–716.

[27] X. Zhang, L. Peng, W. Xu, and L. Kneip, "Accelerating globally optimal consensus maximization in geometric vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[28] T. Huang, Y. Liu, B. Yang, and Y.-H. Liu, "Efficient and globally optimal camera orientation estimation with line correspondences," *IEEE Robotics and Automation Letters*, 2024.

[29] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited,"

in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[30] X. Huang, Y.-J. Huang, Y. Zhang, W. Tian, R. Feng, Y. Zhang, Y. Xie, Y. Li, and L. Zhang, "Open-set image tagging with multi-grained text supervision," *arXiv preprint arXiv:2310.15200*, 2023.
[31] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan *et al.*, "Grounded sam: Assembling open-world models for diverse visual tasks," *arXiv preprint arXiv:2401.14159*, 2024.
[32] I. Suárez, J. M. Buenaposada, and L. Baumela, "Elsed: Enhanced line segment drawing," *Pattern Recognition*, vol. 127, p. 108619, 2022.

APPENDIX I
PROOF FOR THEOREM 1

Recall that Theorem 1 characterizes extreme points of the spherical function $h_2(\vec{\mathbf{u}}|\mathbf{a})$, with the rotation axis $\vec{\mathbf{u}}$ belonging to a sub-cube $\boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}$ in polar coordinates.

Proof: First, rewrite $h_2(\vec{\mathbf{u}}|\mathbf{a})$ as:

$$\begin{aligned} h_2(\vec{\mathbf{u}}|\mathbf{a}) &= \vec{\mathbf{n}}_a^\top [\vec{\mathbf{u}}]_\times^2 \vec{\mathbf{v}}_a = \vec{\mathbf{n}}_a^\top (\vec{\mathbf{u}}\vec{\mathbf{u}}^\top - \mathbf{I}_3)\vec{\mathbf{v}}_a \\ &= \vec{\mathbf{u}}^\top (\vec{\mathbf{n}}_a \vec{\mathbf{v}}_a^\top)\vec{\mathbf{u}} - \vec{\mathbf{n}}_a^\top \vec{\mathbf{v}}_a \\ &= \vec{\mathbf{u}}^\top (\frac{\vec{\mathbf{n}}_a \vec{\mathbf{v}}_a^\top + \vec{\mathbf{v}}_a \vec{\mathbf{n}}_a^\top}{2})\vec{\mathbf{u}} - \vec{\mathbf{n}}_a^\top \vec{\mathbf{v}}_a. \end{aligned}$$

Next, we derive the critical points of $h_2(\vec{\mathbf{u}}|\mathbf{a})$ for $\vec{\mathbf{u}} \in \mathbb{S}^2$. Denote $\mathbf{M}_a := \frac{\vec{\mathbf{n}}_a \vec{\mathbf{v}}_a^\top + \vec{\mathbf{v}}_a \vec{\mathbf{n}}_a^\top}{2}$, and take derivative of $h_2(\vec{\mathbf{u}}|\mathbf{a})$ w.r.t a perturbation $\boldsymbol{\delta} \in \mathbb{R}^3$ on the lie algebra as follows:

$$\begin{aligned} &\frac{\partial h_2(\mathbf{expm}([\boldsymbol{\delta}]_\times)\vec{\mathbf{u}}|\mathbf{a})}{\partial \boldsymbol{\delta}}|_{\boldsymbol{\delta}=0} \\ &= \frac{\partial \left(\mathbf{expm}([\boldsymbol{\delta}]_\times)\vec{\mathbf{u}}\right)^\top \mathbf{M}_a \ \mathbf{expm}([\boldsymbol{\delta}]_\times)\vec{\mathbf{u}}}{\partial \boldsymbol{\delta}}|_{\boldsymbol{\delta}=0}, \\ &= \frac{\partial \vec{\mathbf{u}}^\top (-[\boldsymbol{\delta}]_\times \mathbf{M}_a + \mathbf{M}_a [\boldsymbol{\delta}]_\times)\vec{\mathbf{u}} + o(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}}|_{\boldsymbol{\delta}=0}, \\ &= 2[\vec{\mathbf{u}}]_\times \mathbf{M}_a \vec{\mathbf{u}} = 2\vec{\mathbf{u}} \times (\mathbf{M}_a \vec{\mathbf{u}}), \end{aligned}$$

where $\mathbf{expm}(\cdot)$ is the matrix exponential operation, and we use the equality $[\mathbf{a}]_\times \mathbf{b} = -[\mathbf{b}]_\times \mathbf{a}$ in the last equation. To let the derivative equal zero, one has

$$\vec{\mathbf{u}} \parallel \mathbf{M}_a \vec{\mathbf{u}} \Leftrightarrow \vec{\mathbf{u}} \text{ is an eigenvector of } \mathbf{M}_a.$$

Substitute and verify that $\vec{\mathbf{m}}_a$, $\vec{\mathbf{m}}_a^\perp$, and $\vec{\mathbf{c}}_a$ are three orthogonal eigenvectors of matrix $\mathbf{M}_a$, corresponding to a positive, negative and zero eigenvalue respectively. And $h_2(\vec{\mathbf{u}}|\mathbf{a})$ achieves the global maximum at $\pm\vec{\mathbf{m}}_a$, the global minimum at $\pm\vec{\mathbf{m}}_a^\perp$, and saddle points at $\pm\vec{\mathbf{c}}_a$. Given that $h_2(\vec{\mathbf{u}}|\mathbf{a})$ with $\vec{\mathbf{u}} \in \mathbb{S}^2$ has finite (six) critical points, we conclude that its extreme values for $\vec{\mathbf{u}} \in \boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}$ are achieved either at $\pm\vec{\mathbf{m}}_a$, $\pm\vec{\mathbf{m}}_a^\perp$, or at $\partial\boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}$. ■

We visualize $h_2(\vec{\mathbf{u}}|\mathbf{a})$ on the sphere in Fig. 6.

APPENDIX II
SUPPLEMENTARY RESULTS FOR THEOREM 2

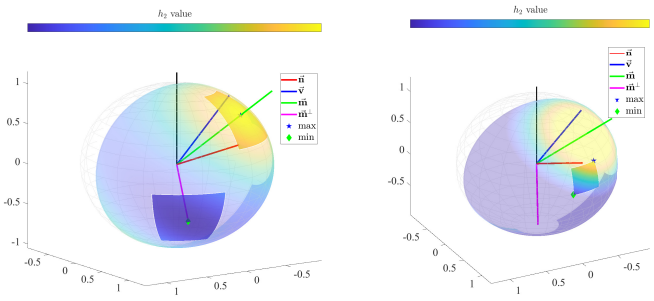
Denote polar coordinates for $\vec{\mathbf{c}}_a := \frac{\vec{\mathbf{v}}_a \times \vec{\mathbf{n}}_a}{\|\vec{\mathbf{v}}_a \times \vec{\mathbf{n}}_a\|}$ as (α_c, ϕ_c):

$$h_1(\vec{\mathbf{u}}|\mathbf{a}) = \sin\alpha_c \sin\alpha \cos(\phi_c - \phi) + \cos\alpha_c \cos\alpha + const.$$

The partial derivative of h_1 with respect to α and ϕ writes

$$\frac{\partial h_1}{\partial \alpha} = \sin\alpha_c \cos\alpha \cos(\phi_c - \phi) - \sin\alpha \cos\alpha_c, \quad (15a)$$

$$\frac{\partial h_1}{\partial \phi} = \sin\alpha_c \sin\alpha \sin(\phi_c - \phi). \quad (15b)$$



(a) values over a half sphere (b) values over a sub-cube

Fig. 6: Landscapes of function $h_2(\vec{\mathbf{u}}|\mathbf{a})$.

For conciseness, we focus on the case where both the sub-cube $\boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}$ and $\vec{\mathbf{c}}_a$ are in the east-hemisphere, i.e., $\phi \in [0, \pi]$ and $\phi_c \in [0, \pi]$. The following arguments can be easily adapted to the other cases. Denote polar coordinates of $\vec{\mathbf{u}}$ which minimizes $h_1(\vec{\mathbf{u}}|\mathbf{a})$ as $\alpha_{\min}$ and $\phi_{\min}$, and of $\vec{\mathbf{u}}$ which maximizes $h_1(\vec{\mathbf{u}}|\mathbf{a})$ as $\alpha_{\max}$ and $\phi_{\max}$. We further denote

$$\alpha_{\text{near}} := \underset{\alpha \in [\alpha_l, \alpha_r]}{\arg\min} |\alpha - \alpha_c|, \quad \phi_{\text{near}} := \underset{\phi \in [\phi_l, \phi_r]}{\arg\min} |\phi - \phi_c|.$$

$$\alpha_{\text{far}} := \underset{\alpha \in [\alpha_l, \alpha_r]}{\arg\max} |\alpha - \alpha_c|, \quad \phi_{\text{far}} := \underset{\phi \in [\phi_l, \phi_r]}{\arg\max} |\phi - \phi_c|.$$

We first give a lemma for $\phi_{\max}$ and $\phi_{\min}$:

Lemma 1: If $(\alpha, \phi) \in \partial\boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}$ is an extreme point for $h_1(\vec{\mathbf{u}}|\mathbf{a})$ on the boundaries of cube, one must have $\phi_{\max} = \phi_{\text{near}}$ and $\phi_{\min} = \phi_{\text{far}}$.

Proof: As the ϕ dimension disappears for $\alpha = 0$ or π, we can focus on $\alpha \in (0, \pi)$. The result naturally arises from two observations on the partial derivative (15b). Firstly, for a fixed $\alpha \in (0, \pi)$, we have $\frac{\partial h_1}{\partial \phi} > 0$ if $\phi < \phi_c$, and $\frac{\partial h_1}{\partial \phi} < 0$ if $\phi > \phi_c$. Secondly, the partial derivative takes the same absolute value for ϕ_1 and ϕ_2 equally distant from ϕ_c. ■

After fixing the value of ϕ according to Lemma 1, we can focus on α by studying (15a).

Lemma 2: Given a fixed value of ϕ and for $\alpha_c \neq \pi/2$, the partial derivative (15a) has a unique zero point $\alpha^* \in [0, \pi]$, and α^* is a **global maximizer** of $h_1(\vec{\mathbf{u}}|\mathbf{a})$ if $|\phi_c - \phi| < \pi/2$, and a **global minimizer** of $h_1(\vec{\mathbf{u}}|\mathbf{a})$ if $|\phi_c - \phi| > \pi/2$.

Proof: Note that (15a) can be organized into a form of $A\sin(\alpha + \bar{\beta})$, with $\bar{\beta}$ a fixed angle. For $\alpha_c \neq \pi/2$, we have $\bar{\beta} \neq 0$ or π, and thus zero point α^* is unique within $[0, \pi]$.

Given $\alpha_c \neq \pi/2$, $\alpha = \pi/2$ is not a zero point of (15a), and thus we can safely rewrite it as:

$$\frac{\partial h_1}{\partial \alpha} = \cos\alpha_c \cos\alpha (\tan\alpha_c \cos(\phi_c - \phi) - \tan\alpha). \quad (16)$$

Based on (16), we list in Table IV four cases for the zero point α^* which completes the proof. ■

Notice that we omit special cases where $\alpha_c = \pi/2$ or $|\phi - \phi_c| = \pi/2$ in Lemma 2 for conciseness, while interested readers can refer to our implementation for details. Based on Lemmas 1 and 2, we propose an efficient procedure to find extreme points of $h_1(\vec{\mathbf{u}}|\mathbf{a})$ on $\partial\boldsymbol{\mathcal{C}}_{\vec{\mathbf{u}}}$ as follows. We introduce

TABLE IV: Four different cases of α_c and $\Delta\phi$

α_c	$\lvert\phi_c - \phi\rvert$	α^*	$\cos\alpha_c \cos\alpha^*$	Extrema
$< \pi/2$	$< \pi/2$	$\in (0, \alpha_c)$	> 0	max
$> \pi/2$	$< \pi/2$	$\in (\alpha_c, \pi)$	> 0	max
$< \pi/2$	$> \pi/2$	$\in (\pi - \alpha_c, \pi)$	< 0	min
$> \pi/2$	$> \pi/2$	$\in (0, \pi - \alpha_c)$	< 0	min

the notation $\alpha^*(\delta\phi)$ for the unique zero point of α given $\delta\phi := |\phi_c - \phi|$, and denote

$$\alpha_{\text{near}}[\alpha^*(\delta\phi)] := \underset{\alpha\in[\alpha_l,\alpha_r]}{\arg\min} |\alpha - \alpha^*(\delta\phi)|,$$

$$\alpha_{\text{far}}[\alpha^*(\delta\phi)] := \underset{\alpha\in[\alpha_l,\alpha_r]}{\arg\max} |\alpha - \alpha^*(\delta\phi)|.$$

Find the maximizer with $\phi = \phi_{\text{near}}$, $\delta\phi_{\text{near}} := |\phi_c - \phi_{\text{near}}|$:

1) If $\delta\phi_{\text{near}} = 0$, we have $\alpha_{\max} = \alpha_{\text{near}}$;
2) if $\delta\phi_{\text{near}} = \pi/2$,

$$\alpha_{\max} = \begin{cases} \alpha_l & \text{if} \quad \alpha_c \le \pi/2, \\ \alpha_r & \text{if} \quad \alpha_c > \pi/2; \end{cases}$$

3) if $\delta\phi_{\text{near}} > \pi/2$, we have $\alpha_{\max} = \alpha_{\text{far}}[\alpha^*(\delta\phi_{\text{near}})]$
4) if $\delta\phi_{\text{near}} < \pi/2$, $\alpha_c < \pi/2$, and $\alpha_l >= \alpha_c$, we have

$$\alpha_{\max} = \alpha_l;$$

5) if $\delta\phi_{\text{near}} < \pi/2$, $\alpha_c > \pi/2$ and $\alpha_r <= \pi - \alpha_c$:

$$\alpha_{\max} = \alpha_r;$$

6) otherwise, $\alpha_{\max} = \alpha_{\text{near}}[\alpha^*(\delta\phi_{\text{near}})]$.

Find the minimizer with $\phi = \phi_{\text{far}}$, $\delta\phi_{\text{far}} := |\phi_c - \phi_{\text{far}}|$:

1) If $\delta\phi_{\text{far}} < \pi/2$, we have $\alpha_{\min} = \alpha_{\text{far}}[\alpha^*(\delta\phi_{\text{far}})]$
2) if $\delta\phi_{\text{near}} = \pi/2$,

$$\alpha_{\min} = \begin{cases} \alpha_r & \text{if} \quad \alpha_c \le \pi/2, \\ \alpha_l & \text{if} \quad \alpha_c > \pi/2; \end{cases}$$

3) if $\delta\phi_{\text{far}} > \pi/2$, $\alpha_c < \pi/2$ and $\alpha_r <= \pi - \alpha_c$:

$$\alpha_{\min} = \alpha_r;$$

4) if $\delta\phi_{\text{far}} > \pi/2$, $\alpha_c > \pi/2$ and $\alpha_l >= \pi - \alpha_c$:

$$\alpha_{\min} = \alpha_l;$$

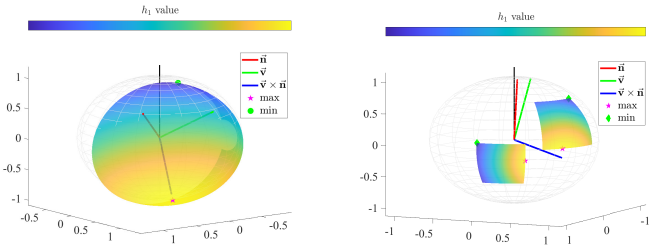
5) otherwise, $\alpha_{\min} = \alpha_{\text{near}}[\alpha^*(\delta\phi_{\text{near}})]$

We finally illustrate Theorem 2 and the above results by visualizing $h_1(\vec{\mathbf{u}}|\mathbf{a})$ on the sphere, as presented in Fig. 7.

APPENDIX III
SEMANTIC LINE MAP CONSTRUCTION

We propose to construct a semantic 3D line map based on posed RGB-D images and their semantic segmentation. The procedure consists of three steps:

1) Extract 2D lines in each image, and assign a semantic label according to segmentation.
2) Inversely project a 2D line l based on depth and regress a 3D line L, assign L with the same semantic as l.
3) After processing all images, cluster and prune the regressed 3D lines.



(a) values over a half sphere (b) values over a sub-cube

Fig. 7: Landscapes of function $h_1(\vec{\mathbf{u}}|\mathbf{a})$.

For the first step, we adopt ELSED [32] to extract 2D lines, and use the rendered semantic mask provided by ScanNet++. For the second step, we dedicate to tackling the background interference issue. As shown in Fig. 8, the inversely projected 3D points can fall on the background while the extracted edge belongs to the foreground object, due to pose or depth error. We propose to evaluate multiple line hypotheses generated from perturbing the 2D line, and select one with a small average depth and mild perturbation.

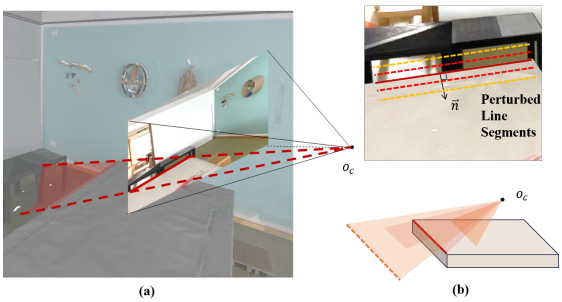


Fig. 8: (a) Regressing a 3D line based on points found by inverse projection is prone to the interference of background points. (b) We select the foreground 3D line from multiple hypotheses obtained from perturbed 2D line segments.

Since a 3D line can be observed by multiple images from different viewpoints, there exist a lot of redundant 3D lines after processing all images. In the last step, we cluster these lines based on geometric constraints. We treat each 3D line $L_m : (\mathbf{p}_m, \vec{\mathbf{v}}_m, s_m)$ as a vertex $\mathcal{V}_m$ on a graph $\mathcal{G}$. For two lines L_m, L_n satisfying the following **parallel** and **proximity** conditions, we connect the two vertices with an edge:

$$\angle(\vec{\mathbf{v}}_m, \vec{\mathbf{v}}_,) < \delta_r \ \textbf{and} \ \|(\mathbf{I}_3 - \vec{\mathbf{v}}_m\vec{\mathbf{v}}_m^\top)(\mathbf{p}_m - \mathbf{p}_,)\| < \delta_t.$$

Denote the set of vertex $\mathcal{V}_m$ and its neighbors as $\mathcal{N}(\mathcal{V}_m)$. We summarize the clustering algorithm in Algorithm 2. The overall procedure is fast and efficient with only several parameters to set, which rarely require tuning across different scenes. In fact, we use the same set of parameters across scenes in experiments. As a demonstration of effectiveness, we present local line maps obtained w/ and w/o the proposed multiple hypothesis and clustering algorithms in Fig. 9.

Algorithm 2 3D Line Clustering

Inputs: graph $\mathcal{G}$, degree threshold δ_d

$\mathcal{V}_{\max}$, $d_{\max} = \textit{Maximal Degree}\ (\mathcal{G})$

while $d_{\max} \geq \delta_d$ **do**
  Remove $\mathcal{N}(\mathcal{V}_{\max})$ from $\mathcal{G}$.
  for each unique label s_m in $\mathcal{N}(\mathcal{V}_{\max})$ **do**
    Register a 3D line $L : (\vec{\mathbf{v}}_{\max}, \mathbf{p}_{\max}, s_m)$.
  end for
end while

(a) (b) (c)

Fig. 9: Regressed 3d lines (a) w/o applying the multi-hypotheses and clustering algorithms, (b) w/o clustering, and (c) with both algorithms applied. Note that Thanks to the multi-hypothesis algorithm, there are much less lines in (b) which fall on the background compared to (a). The clustering algorithm clusters the proximate lines and prune the less-observed lines in (b), giving a much neater map in (c).

APPENDIX IV
FIND BOUNDING FUNCTIONS FOR TRANSLATION PROBLEM IN PNL

For the three degrees of freedom t_x, t_y and t_z in translation, we distinguish the one with largest range, as determined by the environment size. Without generality, we assume t_x is the distinguished parameter. Denote $\mathbf{R}_\sigma^* \vec{\mathbf{n}}_a$ as $\vec{\mathbf{n}}_a^* := (n_{a,x}^*, n_{a,y}^*, n_{a,z}^*)$, and denote $\text{const}_a := \vec{\mathbf{n}}_a^* \bullet \mathbf{p}_a$, we rewrite the residual function $f(t_x, t_y, t_z|\mathbf{a})$ in (8) as follows:

$$t_x n_{a,x}^* + t_y n_{a,y}^* + t_z n_{a,z}^* - \text{const}_a. \tag{17}$$

For t_x in its range $\mathcal{I}_x$ and (t_y, t_z) in a sub-cube $\boldsymbol{\mathcal{C}}_{yz}$, the bounding functions $f_L(t_x)$ and $f_U(t_x)$ for (17) can be easily found by solving a linear programming problem about (t_y, t_z). Take $f_L(t_x)$ as an example. We obtain it by solving

$$\arg\min_{t_y, t_z} \ t_y n_{a,y}^* + t_z n_{a,z}^* \quad (t_y, t_z) \in \boldsymbol{\mathcal{C}}_{yz},$$

and substituting the solution into (17). Since a global optimum of linear programming occurs at the vertices, we simply compare the value achieved by the four vertices of $\boldsymbol{\mathcal{C}}_{yz}$.

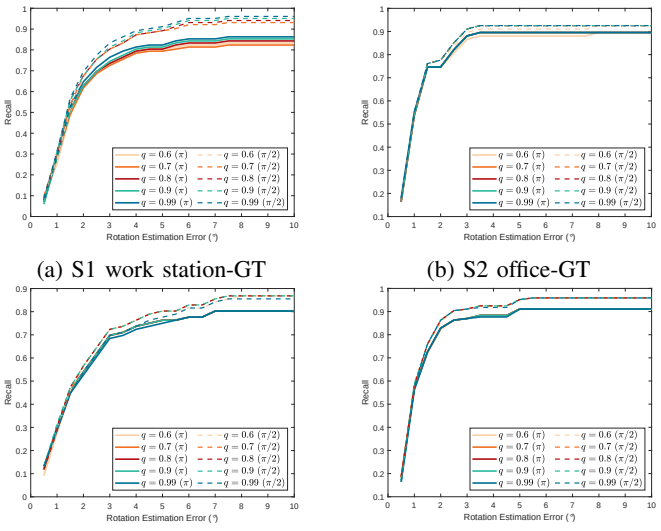
APPENDIX V
SUPPLEMENTARY EXPERIMENT RESULTS

A. Sat-CM v.s. CM in translation estimation estimation

We compare translation estimation accuracy between the Sat-CM and CM method. Since the translation problem (8) is formulated based on a rotation estimate, the orientation accuracy directly governs the precision of translation. In this experiment, we use the most accurate upstream rotation estimates, i.e., those based on the likelihood saturation function (7), to formulate the translation problem. In the legend we use '$\pi-$' and '$\pi/2-$' to distinguish the upstream rotation estimates. We set the translation residual upper bound (used in Assumption 2) at $u_t = 1$, and choose $q = 0.9$ for SCM-GT, $q = 0.5$ for SCM-PR. After solving (8), we improve the output translation candidates with two steps. Firstly, we prune the inlier associations by imposing two physical constraints: the truly-associated 3D line resides in front of the camera and its projection intersects with the image. Next, we fine-tune the translation estimates by minimizing least squares error or the pruned inliers. As present in Table V, Sat-CM performs consistently better than CM across all scenes and settings. Notably, the truncated saturation function performs as good as the likelihood saturation function when using ground truth labels, and presents better robustness under predicted labels by keeping more potentially good candidates for further refinement.

B. parameter sensitivity evaluation

We evaluate performance sensitivity in parameter q by plotting the error curves. For rotation estimation, we choose $q \in \{0.6, 0.7, 0.8, 0.9, 0.99\}$ when using the ground truth labels and choose $q \in \{0.2, 0.35, 0.5, 0.65, 0.8\}$ when using the predicted labels. The results are present in Fig. 10 and 11. For translation estimation, we choose the same set of q values for both ground truth and predicted semantic labels $q \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The results are present in Fig. 12 and Fig. 13.



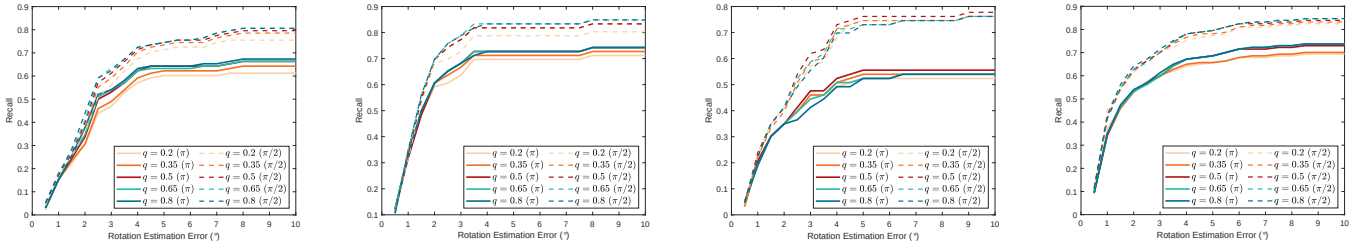
(a) S1 work station-GT (b) S2 office-GT

(c) S3 game bar-GT (d) S4 art room-GT

Fig. 10: Rotation error recall using ground truth semantic labels and different values of q.

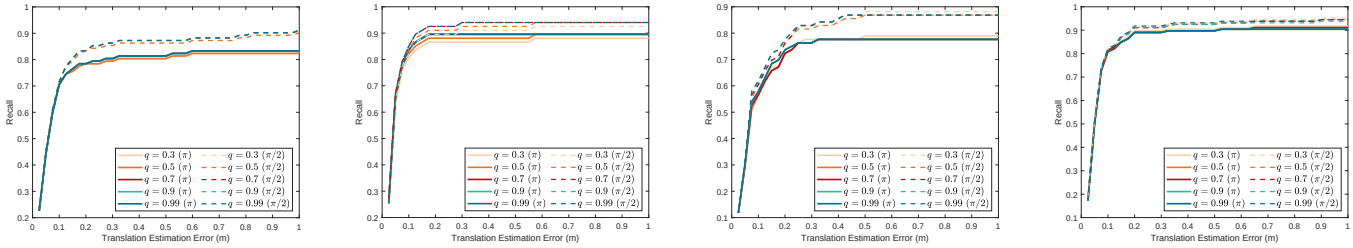
TABLE V: Compare CM with Sat-CM in translation.

Method	S1(workstation)	S2(office)	S3(game bar)	S4(art room)	S1	S2	S3	S4
	25%/50%/75% error quantiles (cm)				Recall at 5cm/10cm/15cm (%)			
π-CM-GT	3.0/5.7/33.9	2.8/4.0/7.9	4.6/9.9/78.4	3.3/5.5/34.9	47/65/72	60/78/81	25/50/61	42/68/71
π-SCM0-GT	2.9/5.9/11.0	2.4/3.8/7.1	4.0/7.1/22.2	2.9/5.0/8.5	45/73/80	63/84/88	30/58/68	50/77/84
π-SCM1-GT	3.0/5.8/14.1	2.5/4.0/6.7	4.0/6.8/22.6	3.0/5.0/8.2	44/71/76	66/84/88	30/58/68	50/82/85
$\pi/2$-CM-GT	3.0/5.5/15.6	2.8/4.0/7.9	4.2/8.6/28.1	3.3/5.5/19.7	48/66/75	60/78/82	26/51/63	42/68/71
$\pi/2$-SCM0-GT	2.8/5.9/10.8	2.4/3.8/6.7	4.0/6.9/19.4	2.9/4.9/8.2	46/74/85	64/85/91	32/61/72	51/78/86
$\pi/2$-SCM1-GT	2.7/5.6/11.3	2.5/3.8/6.6	4.0/6.5/18.9	3.0/4.9/8.0	45/72/80	67/85/91	32/62/72	51/82/88
π-CM-PR	5.2/77.4/198.4	5.2/63.0/214.9	8.2/129.1/241.2	6.9/121.5/226.9	24/37/42	23/41/44	17/27/35	18/29/34
π-SCM0-PR	4.8/13.3/150.7	3.7/7.7/197.5	7.2/100.7/232.0	5.9/21.2/195.5	28/44/53	35/59/62	17/29/37	20/39/45
π-SCM1-PR	4.5/13.8/150.7	3.7/7.3/194.5	6.4/135.8/226.6	5.9/33.8/208.3	28/44/53	33/58/61	19/33/41	20/38/45
$\pi/2$-CM-PR	5.2/16.7/175.2	4.7/11.8/194.5	7.7/61.1/234.9	6.4/107.0/208.1	24/42/48	26/47/52	17/29/37	19/31/36
$\pi/2$-SCM0-PR	4.4/10.6/90.1	3.7/6.8/49.2	6.6/32.3/199.0	5.5/14.4/119.8	30/49/60	36/65/70	19/33/43	23/43/51
$\pi/2$-SCM1-PR	4.5/10.2/113.2	3.6/6.3/105.9	5.2/26.7/224.5	5.7/18.0/192.1	29/49/59	36/65/70	24/40/49	23/41/48



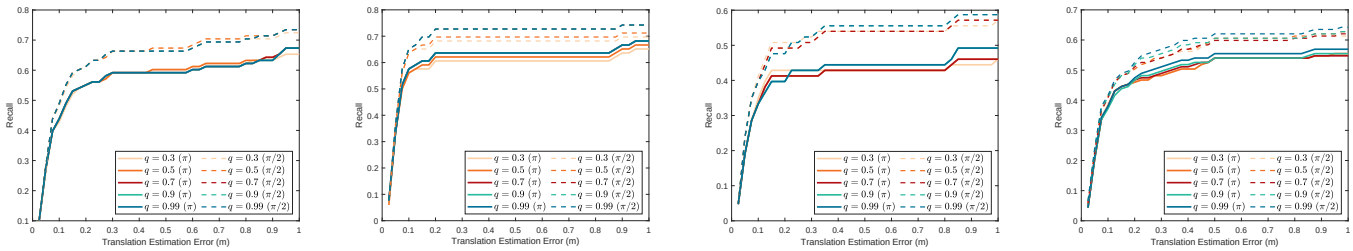
(a) S1 work station-PR (b) S2 office-PR (c) S3 game bar-PR (d) S4 art room-PR

Fig. 11: Rotation error recall using predicted semantic labels and different values of q.



(a) S1 work station-GT (b) S2 office-GT (c) S3 game bar-GT (d) S4 art room-GT

Fig. 12: Translation error recall using ground truth semantic labels and different values of q.



(a) S1 work station-PR (b) S2 office-PR (c) S3 game bar-PR (d) S4 art room-PR

Fig. 13: Translation error recall using predicted labels and different values of q.