ActLoc: Learning to Localize on the Move via Active Viewpoint Selection

Jiajie Li[1,*], Boyang Sun[1,*], Luca Di Giammarino[2], Hermann Blum[1,4], Marc Pollefeys[1,3]
[1]ETH Zürich [2]Sapienza University of Rome [3]Microsoft [4]University of Bonn
[*]Equal contribution
https://boysun045.github.io/ActLoc-Project/

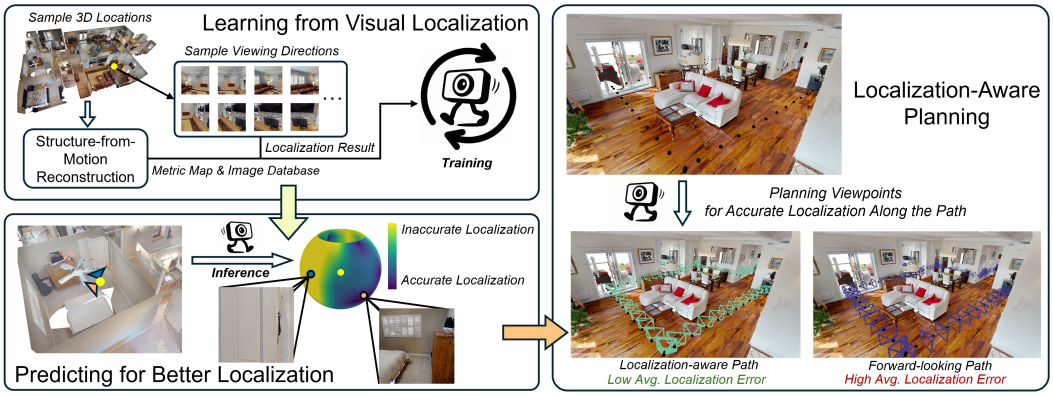

Figure 1: **ActLoc** is an active viewpoint selection framework designed to improve visual localization for robot navigation. **Left**: The model is trained using supervision from Visual Localization, learning to predict distributions of localization accuracy over yaw and pitch angles at arbitrary 3D locations. **Right**: The trained model is integrated into the path planning process, enabling the robot to select viewpoints at each waypoint that increase localization robustness along its trajectory.

Abstract: Reliable localization is critical for robot navigation, yet most existing systems implicitly assume that all viewing directions at a location are equally informative. In practice, localization becomes unreliable when the robot observes unmapped, ambiguous, or uninformative regions. To address this, we present **ActLoc**, an **act**ive viewpoint-aware planning framework for enhancing **loc**alization accuracy for general robot navigation tasks. At its core, ActLoc employs a large-scale trained attention-based model for viewpoint selection. The model encodes a metric map and the camera poses used during map construction, and predicts localization accuracy across yaw and pitch directions at arbitrary 3D locations. These per-point accuracy distributions are incorporated into a path planner, enabling the robot to actively select camera orientations that maximize localization robustness while respecting task and motion constraints. ActLoc achieves state-of-the-art results on single-viewpoint selection and generalizes effectively to full-trajectory planning. Its modular design makes it readily applicable to diverse robot navigation and inspection tasks.

Keywords: Active Vision; Robot Navigation; Visual Localization

1 Introduction

In recent years, visual localization has gained significant traction in robotics for two main reasons. First, compared to LiDAR, cameras are far more affordable, reducing sensor costs and making

robotic systems more accessible. Second, visual localization extends beyond robotics, being widely adopted in handheld and head-mounted devices where LiDAR sensors are often too bulky and heavy. Visual maps can also be shared across devices, enabling multi-agent spatial alignment [1, 2, 3] and facilitating human–robot collaboration [4, 5].

Despite these advantages, visual localization remains more challenging than LiDAR-based positioning [6, 7, 8]. For example, in the Hilti SLAM Challenge 2023 [9], the vision-only track achieved barely half the scores of LiDAR-based methods. A key limitation stems from the narrow field of view (FoV) of cameras, which prevents omnidirectional sensing readily available with more expensive sensor setups [10, 11]. Since visual localization relies heavily on image features, viewpoints covering feature-rich regions tend to yield more reliable results. This observation has motivated active localization strategies [12, 13, 14], where viewpoint selection is integrated into the robot's motion planning. By actively adjusting its position and orientation, a robot can track feature-rich regions and improve localization accuracy along its trajectory.

Active vision methods have progressed from analytical modeling [12, 15] to deep learning–based approaches [13, 14, 16, 17]. In the context of active localization, learning-based methods improve accuracy but remain computationally expensive, since evaluating localization quality at each viewpoint demands substantial resources—an issue for real-time systems where fast decisions are essential. Moreover, these methods are trained on small datasets, limiting their generalization ability. Finally, most approaches assess viewpoints independently, requiring sparse sampling and decoupling viewpoint selection from the path planning process.

To address these challenges, we propose an efficient and scalable approach. Instead of evaluating viewpoints individually, our model encodes scene information and directly predicts a spherical distribution of localization quality at each 3D waypoint. This allows dense estimation across multiple yaw–pitch directions and supports trajectory optimization that jointly accounts for task objectives and localization robustness. The resulting system is both computationally efficient and suitable for continuous real-time navigation.

In summary, our main contributions are:

- A learning-based model trained on visual localization tasks that predicts a distribution of localization accuracy over multiple yaw–pitch directions in a single forward pass.
- **ActLoc**, an active localization planner that integrates the model into a complete navigation system for online viewpoint selection and trajectory planning.
- Comprehensive experiments across multiple datasets, evaluating both single-viewpoint selection and long-horizon path planning, where ActLoc consistently outperforms baselines and heuristic methods.

2 Related Work

2.1 Visual Localization

Visual localization refers to estimating the 6-DoF camera pose of a query image within a known environment. Existing approaches can be broadly divided into two categories. The first is the two-step approach, which establishes correspondences between the query and a database of reference images, followed by pose estimation through geometric optimization. These correspondences may be based on sparse visual features [18, 19, 20, 21, 22, 23, 24, 25] or dense pixel-level matches [26, 27, 28, 29, 30, 31, 32]. The second is the direct approach, also known as pose regression, which estimates the camera pose directly from the input image, typically using learning-based models [33, 34, 35, 36, 37, 38, 39, 40, 41]. While both approaches have achieved strong results, they remain inherently *passive*: the camera pose is inferred solely from the captured view, without reasoning about which viewpoint would yield the most reliable localization. This limitation has motivated research into *active visual localization*.

2.2 Active Viewpoint Selection

Active visual localization extends beyond passive pose estimation by allowing an agent to actively select its viewpoints in order to improve localization reliability. This direction builds upon the broader field of active perception, which has long studied strategies for guiding sensor orientations and robot motion to maximize the informativeness of observations [42, 43, 44, 45, 46, 47, 48]. By integrating perception with planning, these approaches enable robots not only to gather richer visual cues but also to make decisions that directly impact navigation performance.

Early studies on active localization focused on designing evaluation metrics to identify informative viewpoints. Such metrics are typically geometric, for instance using Fisher information to quantify the visibility and spatial distribution of landmarks [15]. While conceptually interpretable, these hand-crafted metrics are often environment-specific and become computationally expensive when extended to dense viewpoint sampling across large maps.

To overcome the limitations of hand-crafted criteria, recent work has turned to learning-based formulations. One line of work treats pose estimation as a regression problem using convolutional networks [49, 50]. Another line integrates perceptual models with policy learning to guide localization [51, 52, 53], sometimes augmented with higher-level cues such as semantics [54]. While these policy-driven methods enable adaptive viewpoint control, they often rely on reinforcement learning, which couples perception tightly with action execution and hinders generalization. A recent alternative formulates viewpoint selection as a classification task [13], yielding promising performance but relying on user-annotated supervision and remaining limited to per-location viewpoint decisions without trajectory-level planning. The most relevant recent advance is Learning-Where-to-Look LWL) [14], which represents the state-of-the-art in active localization. LWL trains a lightweight multi-layer perceptron (MLP) to classify viewpoints as suitable or unsuitable for localization, with training labels generated in a self-supervised way from voxel-grid–based sampling and visibility checks. At test time, however, it requires dense viewpoint sampling and extensive preprocessing (including landmark visibility evaluation and feature binning), making inference computationally costly even with GPU acceleration. Although LWL presents qualitative planning results, it does not offer a fully integrated or quantitatively evaluated trajectory-level planning system.

These limitations motivate our approach, which removes voxel-grid constraints, predicts localization performance across multiple viewpoints in a single pass, and integrates these predictions into motion planning to enable active viewpoint selection along a trajectory.

3 Method

3.1 System Overview

Our system consists of two main components: (1) an attention-based model trained on Structure-from-Motion (SfM) data to predict visual localization performance over local yaw and pitch angles. We denote this representation as *LocMap*. (2) The integration of LocMap into a path planning pipeline, which, given a sequence of *waypoints* (robot positions in 3D space), actively selects *viewpoints* (camera orientations at each waypoint) to improve localization accuracy while respecting motion constraints such as continuity. ActLoc predicts the localization outcome for multiple candidate viewpoints in a single forward pass and combines these predictions with planning costs to balance accuracy and feasibility. Figure 2 provides an overview of the full pipeline.

3.2 Predicting Localization Performance

We define active localization as the problem of selecting viewpoints along a given sequence of waypoints such that the robot achieves accurate visual localization while following the trajectory. At each waypoint, the planner chooses a viewpoint so that images captured during execution yield reliable localization results. We assume that the environment has been previously mapped, which is

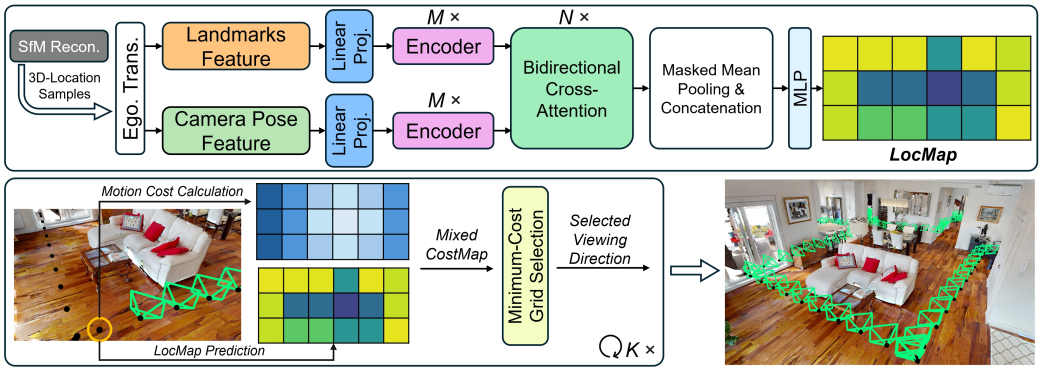

Figure 2: **System Overview. Top**: The core of ActLoc is an attention-based network that processes multi-modal inputs from an SfM model, including the metric map and camera poses. The network predicts localization performance as a two-dimensional grid over multiple pitch and yaw rotations. **Bottom**: During path planning, ActLoc constructs a mixed cost map at each candidate location by combining the network prediction with additional factors, such as motion constraints, and selects the viewing direction that minimizes this cost.

standard in visual localization since methods typically query an existing database. This assumption naturally applies to many robotics tasks, such as inspection, monitoring, and indoor service.

To select a viewpoint at a waypoint, the robot must leverage prior knowledge of the scene from mapping and adapt it to the specific location. We therefore train a model that encodes this prior knowledge and predicts the expected localization performance across possible viewing directions. As shown in Figure 2, the model outputs the *LocMap*, a two-dimensional grid of size $H \times W$, where each cell represents the predicted localization performance for a candidate viewpoint. The horizontal axis corresponds to yaw rotations, and the vertical axis corresponds to pitch rotations. We exclude extreme pitch angles that point toward the ceiling or floor, as well as in-plane rotations around the camera's principal axis, since these provide negligible benefit for visual localization [55]. Experiments in Appendix B.3 further confirm that such rotations often degrade localization performance.

Unlike prior work that either scores only the best viewpoint [15] or evaluates multiple viewpoints independently without modeling correlations across orientations [13, 14], our model predicts the entire distribution of viewpoint performance in a single forward pass. This design reflects the natural fact that multiple viewpoints may provide good localization, while also avoiding redundant computation from repeated single-viewpoint inference. Moreover, modeling the joint distribution allows the network to capture spatial correlations between viewpoints, which is crucial for accurately predicting localization performance at a waypoint.

We propose an attention-based model to predict the LocMap. This model continuously queries a SfM pipeline to generate training data. Most SfM and visual localization approaches maintain a reconstruction database that includes a metric map and a set of camera poses used for reconstruction. We select these two components as the raw inputs to our pipeline: the 3D landmarks from the metric map $\mathbf{P} \in \mathbb{R}^{3\times N}$ and the camera poses $\mathbf{T} \in \mathbb{SE}(3)^M$. For a given waypoint $\mathbf{x} \in \mathbb{R}^3$, our model predicts the LocMap:

$$\mathbf{C}_{\mathbf{x}}^{H\times W} = f_\theta(\mathbf{P}, \mathbf{T}, \mathbf{x}) \tag{1}$$

where $\mathbf{C}_{\mathbf{x}}$ denotes the predicted distribution of localization performance across candidate viewpoints at $\mathbf{x}$. Each cell stores the likelihood that the localization error at the corresponding yaw–pitch direction falls below a threshold. This allows the model to reason about all orientations jointly, instead of evaluating them one by one.

3.3 Model Architecture

Our model takes multi-modal input from landmarks and camera poses, encoding each with self-attention layers and fusing them through cross-attention. Given the raw inputs $\mathbf{T}$, $\mathbf{P}$, and $\mathbf{x}$, we apply several preprocessing steps. First, we remove highly uncertain 3D landmarks from $\mathbf{P}$, following [14]. Next, we apply an egocentric transformation that shifts $\mathbf{T}$ and the filtered $\mathbf{P}$ from the world frame to the local frame of $\mathbf{x}$. Importantly, we shift only their origins while preserving rotations to maintain consistency with the default viewing direction at $\mathbf{x}$. This implicitly incorporates waypoint location into the inputs without additional modules. Finally, we crop the filtered $\mathbf{P}$ within a bounding box centered at $\mathbf{x}$, enforcing locality and normalizing input scale. The full preprocessing pipeline is shown in Figure 5.

After preprocessing, camera features are constructed from $\mathbf{T}$ by extracting their Euclidean representation in $\mathbb{R}^7$, consisting of the camera center coordinates and a unit quaternion for orientation $(\mathbf{p_c}, \mathbf{q_c})$. Landmark features are built from the filtered $\mathbf{P}$, with each landmark represented as a 6D vector containing 3D coordinates and normalized RGB values.

Both feature types are independently projected into a shared high-dimensional embedding space using linear layers. Transformer encoder blocks are then applied to each modality, extracting context-aware representations. To integrate modalities, we employ a multi-layer bidirectional cross-attention block, allowing pose embeddings to attend to landmark embeddings and vice versa. This design enables bidirectional information flow and produces fused features that capture the interplay between sparse camera poses and dense 3D landmarks.

Because attention operates on sequences, we apply masked mean pooling to aggregate features into fixed-size vectors while ignoring padding introduced during batching. This yields one vector summarizing pose information and another summarizing landmarks. These are concatenated and passed through an MLP to produce the final output logits. For efficiency, all attention layers use *FlashAttentionV2* [56], which reduces memory complexity from $O(N^2)$ to $O(N)$ with respect to sequence length and provides substantial speedups, especially for long sequences.

3.4 Training Data Creation

We train our model entirely on simulated data derived from the HM3D dataset [57]. Specifically, we use the first 100 single-floor scenes, with 90 for training/validation and 10 reserved for testing.

To construct the prior scene maps, we first sample free-space points at typical operation heights between 1.5 and 2.0 meters. At each point, the camera performs a full 360° sweep in 36° azimuth increments. For each orientation, an integer elevation angle is randomly sampled from $[-15^\circ, 15^\circ)$, and an image is captured. We then run SfM on these images using SuperPoint [55] features with the Hierarchical Localization (hloc) framework [23] to obtain 3D landmarks.

To generate training samples, we independently sample waypoints within each scene at heights between 0.4 and 2.0 meters, making our approach free of grid-based assumptions as in [14]. At each waypoint, the camera rotates horizontally at 20° intervals covering the full $[-180^\circ, 180^\circ)$ range. For each azimuth, we additionally vary the elevation from $[-60^\circ, 60^\circ)$ at 20° intervals. Finally, we apply hloc and COLMAP [58] to localize each viewpoint against the metric map (3D landmarks), yielding ground-truth localization performance. The full data generation pipeline is illustrated in Figure 4.

3.5 Path Planning with Localization Awareness

LocMap provides a structured representation for viewpoint selection in both single- and multi-viewpoint planning. Unlike prior work that evaluates viewpoints independently, LocMap predicts the distribution of localization performance across all viewpoints at a waypoint in a single inference. This enables optimizing viewpoint choices along a waypoint sequence to achieve higher overall localization accuracy during trajectory execution.

For a given waypoint $\mathbf{x}$, LocMap $\mathbf{C_x}$ provides a direct selection criterion: choosing the grid cell with the highest predicted localization performance. However, during path execution the robot must also satisfy motion constraints, such as limiting rotations between consecutive frames. To address this, we define a mixed cost map that combines localization performance with motion smoothness:

$$\mathbf{M_x} = \mathbf{C_x} + \lambda \cdot \mathbf{D_x} \tag{2}$$

where $\mathbf{D_x}$ is the Mahalanobis distance from each candidate viewpoint to the previously selected viewpoint, and λ balances the trade-off between localization accuracy and smooth motion.

As the robot follows the waypoint sequence, it computes $\mathbf{M_x}$ at each step and selects the viewpoint with the minimum cost. This strategy maintains a balance between accurate localization and motion feasibility, and can be extended to incorporate additional task-specific metrics.

4 Experiments

We design experiments to address the following questions: (a) How well does our model perform in individual viewpoint selection? (b) When integrated into ActLoc, how effectively does it balance localization performance with motion constraints? (c) How adaptable is ActLoc to different input modality configurations, and how do these configurations perform?

To evaluate (a), we use test scenes from HM3D and ScanNet [59] and compare ActLoc against two baselines: FIF [15] and LWL [14]. **FIF** is an information-theoretic method based on Fisher Information, which models the camera as a bearing vector to make the measure rotation-invariant. Localization quality is estimated using Gaussian Process regression and standard information-theoretic metrics, including minimum eigenvalue, determinant, and trace; we report the best-performing metric. **LWL**, in contrast, employs a lightweight MLP with visibility checks and image binning to generate fixed-length inputs for classifying whether a viewpoint is suitable for localization. We trained LWL on the same dataset for fairness.

To evaluate (b), we integrate the model into the planning pipeline and test it on five HM3D scenes. To better approximate real-world conditions, we sparsify the reconstructed models of these scenes. We compare our approach against LWL and a heuristic forward-facing planner. All experiments are conducted on a workstation with a single NVIDIA GeForce RTX 4090 GPU.

4.1 Single-viewpoint Selection

For the single-viewpoint selection task, we follow the protocol of [14], reporting the percentage of waypoints localized within four thresholds: 0.1m/1°, 0.25m/2°, 0.5m/5°, and 5m/10°. We also report a theoretical upper bound, counting a waypoint as successfully localized if any of its candidate viewpoints meets the threshold.

Our model is configured to output a 6×18 LocMap, where each cell corresponds to a candidate viewpoint. We adopt this resolution as a trade-off between efficiency and accuracy (see Appendix B.7). We select the viewpoint with the highest probability of being well-localized. For a fair comparison, LWL is evaluated over the same 6×18 set of viewpoints, with the best-scoring direction chosen.

The results in Table 1 show that our model consistently outperforms both baselines at all thresholds. This is particularly noteworthy because our model predicts localization performance across multiple viewpoints simultaneously, which is a more challenging task than the single-viewpoint formulation in LWL, yet it still achieves higher accuracy. These results highlight the model's ability to capture global spatial structure from SfM inputs and its robustness in generalizing to unseen scenes.

Efficiency is another key advantage. The average processing time per waypoint is 110 ms for our method, compared to 8230 ms for LWL (batch mode). This efficiency gap grows further with higher-resolution LocMaps or large-scale scenes, as shown in Appendix B.4 and Appendix B.5. These results highlight the practicality of our approach for real-time integration into planning pipelines. Additional results in Appendix B.1 further show that ActLoc remains robust under various levels of reconstruction sparsity.

	Method	0.1m/1°	0.25m/2°	0.5m/5°	5m/10°
HM3D	LWL [14]	90.04	90.98	91.17	91.92
	FIF [15]	60.71	62.41	63.53	64.66
	ActLoc	**92.11**	**92.48**	**92.86**	**93.23**
	Upper Bound	97.74	97.74	97.93	98.50
Scannet	LWL [14]	88.89	89.90	89.90	92.93
	ActLoc	**91.92**	**94.95**	**96.97**	**96.97**
	Upper Bound	100.00	100.00	100.00	100.00

Table 1: **Single-viewpoint Selection Results.** Localization success rates (%) on ten HM3D and five ScanNet scenes. Metrics follow the Long-Term Visual Localization benchmark [60], using three standard thresholds and one additional fine-grained indoor threshold (similar to [14]). Our ActLoc model achieves the best results. Best results are highlighted in bold.

4.2 Path Planning Evaluation

For evaluation in a planning setting, we select several key locations in each scene to ensure coverage of most regions. Path planning is first performed with OMPL [61] to obtain waypoint sequences, which are then interpolated to generate dense waypoints at 20 cm intervals. Viewpoint selection is performed using ActLoc, LWL, and a forward-facing baseline. As in the single-viewpoint evaluation, visual localization is executed with hloc using the selected viewpoints. For ActLoc, we tune λ in the mixed cost map (Eq. 2). And we restrict sampling in LWL to a local neighborhood around the previously selected viewpoint.

We evaluate performance using two metrics: success rate and average error across successful localizations. A localization is considered successful if the error is below 5 m / 10°. Results are summarized in Table 2. Across all scenes, ActLoc achieves the highest success rate and, in most cases, the lowest average translation and rotation error. Importantly, since LWL requires evaluating every frame, we equalize processing times between ActLoc and LWL for fairness. Under this constraint, LWL is often unable to sample sufficient candidates, resulting in degraded performance.

Figure 3 shows qualitative examples of viewpoint selection along waypoint sequences. Thanks to the mixed cost map, ActLoc not only improves localization performance but also maintains smooth rotations between consecutive frames, respecting the robot's kinematic constraints. These results demonstrate that ActLoc integrates seamlessly into path planning, balancing localization accuracy with motion feasibility while remaining computationally efficient for real-time deployment.

Scene ID	Forward	LWL [14]	ActLoc
00005	40.26 / 0.09m / 1.21°	36.25 / 0.10m / 1.20°	**52.75** / 0.07m / 1.09°
00080	11.03 / 0.19m / 1.17°	3.23 / 0.24m / 1.57°	**24.00** / 0.05m / 0.80°
00105	20.10 / 0.22m / 2.34°	10.54 / 1.13m / 3.78°	**27.25** / 0.16m / 1.92°
00110	12.45 / 0.20m / 2.13°	31.40 / 0.09m / 1.71°	**35.95** / 0.07m / 1.37°
00254	29.61 / 0.13m / 1.27°	27.11 / 0.08m / 1.43°	**46.08** / 0.14m / 1.62°

Table 2: **Quantitative Planning Results.** *Success Rate (%)/ Translation Error(m) / Rotation Error(°)* across five HM3D test scenes. ActLoc consistently achieves the highest success rate and often the lowest errors, demonstrating its effectiveness in planning settings.

4.3 Ablation Study

We performed an ablation study to evaluate the individual contributions of camera pose and metric map inputs. Two single-modality variants were tested: ActLoc-Cam, which uses only camera poses, and ActLoc-Map, which uses only point clouds. Both were trained on the same HM3D scenes [57] as the full model.

Method	0.1m/1°	0.25m/2°	0.5m/5°	5m/10°
ActLoc-Cam	84.96	86.65	87.97	88.16
ActLoc-Map	89.10	89.85	90.04	90.41
ActLoc-Full	**92.11**	**92.48**	**92.86**	**93.23**

Table 3: **Ablation Study.** Localization success rates (%) at different thresholds for ActLoc-Full, ActLoc-Cam (camera poses only), and ActLoc-Map (point cloud only). ActLoc-Full achieves the best results across all thresholds, confirming the complementary contributions of both modalities.

For ActLoc-Map, the camera pose branch and cross-attention blocks were removed, retaining only the landmarks encoder and classifier. ActLoc-Cam used a corresponding simplified architecture focused solely on camera poses.

As shown in Table 3, ActLoc-Cam achieved the lowest accuracy, while ActLoc-Map performed better but still fell short of the full model. ActLoc-Full outperformed both across all thresholds, confirming the complementary nature of the two modalities and validating the benefit of joint fusion for active viewpoint selection.

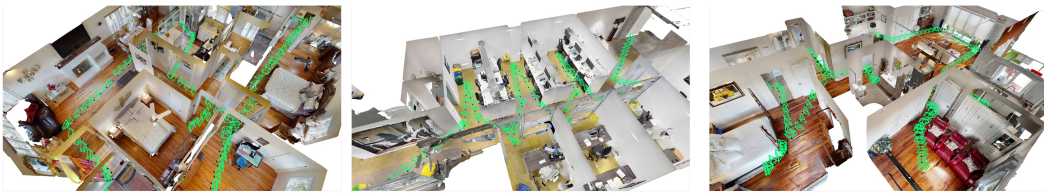

Figure 3: **Qualitative Planning Results.** Visualization of ActLoc's viewpoint selection along waypoint sequences across different test scenes. The planned viewpoints demonstrate how ActLoc adapts camera orientation dynamically to balance localization performance and motion smoothness.

5 Conclusion

In this work, we introduced ActLoc, an active viewpoint-aware planning framework aimed at enhancing localization accuracy for general robot navigation tasks. ActLoc leverages an attention-based model that predicts localization performance over a dense grid of pitch and yaw angles for arbitrary 3D waypoints, enabling robust viewpoint selection. By integrating these predictions into the path planning process, ActLoc allows robots to dynamically adjust camera orientations to maximize localization robustness while respecting motion and task constraints.

Our experiments demonstrated that ActLoc achieves state-of-the-art performance in single-viewpoint selection and generalizes effectively to full-trajectory planning. Notably, ActLoc provides significant runtime improvements over existing methods, making it well-suited for real-time deployment in practical navigation scenarios.

While our approach shows strong performance, further work is needed to enhance generalization across different feature sets and environments. Future research could explore incorporating data augmentation, normalization strategies, or self-supervised learning to improve robustness. Additionally, extending ActLoc to multi-modal sensor fusion or adapting it for dynamic environments could broaden its applicability to more complex, real-world robotics tasks.

6 Limitations

The design of ActLoc naturally supports extensions to multi-class discretizations or continuous regression for finer-grained estimation. However, our experiments (Table 5) show that a naive multi-class formulation with the same training setup does not improve single-viewpoint selection performance. We hypothesize that such extensions may require additional system design and training on larger-scale datasets. At the same time, ActLoc-Multi already provides utility by generating global score heatmaps that reveal spatial variations in localization difficulty (Appendix C.3). This suggests that finer-grained outputs may still be valuable for higher-level planning tasks, even if they do not directly improve single-viewpoint selection.

We also observe limitations in cross-domain generalization. When using alternative visual features, such as SIFT, ActLoc does not outperform LWL [14], as shown in Table 9. This highlights a trade-off in our design: while processing raw inputs in a single forward inference enables high efficiency, it may reduce robustness to different feature types. Future work could address this by exploring input normalization or data augmentation strategies.

Acknowledgments

We sincerely thank the reviewers for their thoughtful comments and constructive suggestions. Their detailed feedback helped us clarify the presentation, improve the organization, and better highlight the contributions of this work. This work has been partially supported by PNRR MUR project PE0000013-FAIR.

References

[1] A. Cramariuc, L. Bernreiter, F. Tschopp, M. Fehr, V. Reijgwart, J. Nieto, R. Siegwart, and C. Cadena. maplab 2.0 – A Modular and Multi-Modal Mapping Framework. *IEEE Robotics and Automation Letters*, 8(2):520–527, 2023. doi:10.1109/LRA.2022.3227865.

[2] Y. Chang, Y. Tian, J. P. How, and L. Carlone. Kimera-multi: a system for distributed multi-robot metric-semantic simultaneous localization and mapping. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11210–11218, 2021. doi:10.1109/ICRA48506.2021.9561090.

[3] H. Blum, A. Mercurio, J. O'Reilley, T. Engelbracht, M. Dusmanu, M. Pollefeys, and Z. Bauer. Crocodl: Cross-device collaborative dataset for localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[4] J. Chen, B. Sun, M. Pollefeys, and H. Blum. A 3d mixed reality interface for human-robot teaming. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11327–11333. IEEE, 2024.

[5] S. A. Salunkhe, P. Nedunghat, L. Morando, N. Bobbili, G. Li, and G. Loianno. Intuitive human-drone collaborative navigation in unknown environments through mixed reality. In *2025 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 862–868. IEEE, 2025.

[6] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette. Ct-icp: Real-time elastic lidar odometry with loop closure. pages 5580–5586. IEEE, 2022.

[7] S. Ferrari, L. Di Giammarino, L. Brizi, and G. Grisetti. Mad-icp: It is all about matching data–robust and informed lidar odometry. 2024.

[8] Y. Pan, X. Zhong, L. Wiesmann, T. Posewsky, J. Behley, and C. Stachniss. PIN-SLAM: LiDAR SLAM Using a Point-Based Implicit Neural Representation for Achieving Global Map Consistency. *IEEE Transactions on Robotics (TRO)*, 40:4045–4064, 2024. URL https://www.ipb.uni-bonn.de/wp-content/papercite-data/pdf/pan2024tro.pdf.

[9] A. D. Nair, J. Kindle, P. Levchev, and D. Scaramuzza. Hilti slam challenge 2023: Benchmarking single+ multi-session slam across sensor constellations in construction. 2024.

[10] L. Di Giammarino, L. Brizi, T. Guadagnino, C. Stachniss, and G. Grisetti. Md-slam: Multi-cue direct slam. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11047–11054. IEEE, 2022.

[11] K. Ćwian, L. Di Giammarino, S. Ferrari, T. Ciarfuglia, G. Grisetti, and P. Skrzypczyński. Madba: 3d lidar bundle adjustment–from uncertainty modelling to structure optimization. *IEEE Robotics and Automation Letters*, 2025.

[12] J. Lim, N. Lawrance, F. Achermann, T. Stastny, R. Bähnemann, and R. Siegwart. Fisher information based active planning for aerial photogrammetry. pages 1249–1255. IEEE, 2023.

[13] M. Hanlon, B. Sun, M. Pollefeys, and H. Blum. Active visual localization for multi-agent collaboration: A data-driven approach. IEEE, 2024.

[14] L. Di Giammarino, B. Sun, G. Grisetti, M. Pollefeys, H. Blum, and D. Barath. Learning where to look: Self-supervised viewpoint selection for active localization using geometrical information. pages 188–205. Springer, 2024.

[15] Z. Zhang and D. Scaramuzza. Beyond point clouds: Fisher information field for active visual localization. pages 5986–5992. IEEE, 2019.

[16] L. Bartolomei, L. Teixeira, and M. Chli. Semantic-aware active perception for uavs using deep reinforcement learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3101–3108. IEEE, 2021.

[17] H. Hu, S. Pan, L. Jin, M. Popović, and M. Bennewitz. Active implicit reconstruction using one-shot view planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12477–12483. IEEE, 2024.

[18] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. pages 667–674. IEEE, 2011.

[19] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. pages 752–765. Springer, 2012.

[20] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. 39(9):1744–1756, 2016.

[21] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla. Are large-scale 3d models really necessary for accurate visual localization? pages 1637–1646, 2017.

[22] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-net: A trainable cnn for joint description and detection of local features. pages 8092–8101, 2019.

[23] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. pages 12716–12725, 2019.

[24] 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. pages 3247–3257, 2021.

[25] V. Panek, Z. Kukelova, and T. Sattler. Meshloc: Mesh-based visual localization. pages 589–609. Springer, 2022.

[26] E. Brachmann and C. Rother. Learning less is more-6d camera localization via 3d surface regression. pages 4654–4662, 2018.

[27] E. Brachmann and C. Rother. Expert sample consensus applied to camera re-localization. pages 7525–7534, 2019.

[28] L. Yang, Z. Bai, C. Tang, H. Li, Y. Furukawa, and P. Tan. Sanet: Scene agnostic network for camera localization. pages 42–51, 2019.

[29] T. Cavallari, S. Golodetz, N. A. Lord, J. Valentin, V. A. Prisacariu, L. Di Stefano, and P. H. Torr. Real-time rgb-d camera pose estimation in novel scenes using a relocalisation cascade. 42(10):2465–2477, 2019.

[30] X. Li, S. Wang, Y. Zhao, J. Verbeek, and J. Kannala. Hierarchical scene coordinate classification and regression for visual localization. pages 11983–11992, 2020.

[31] S. Tang, C. Tang, R. Huang, S. Zhu, and P. Tan. Learning camera localization via dense scene matching. pages 1831–1841, 2021.

[32] S. Dong, S. Wang, Y. Zhuang, J. Kannala, M. Pollefeys, and B. Chen. Visual localization via few-shot scene region classification. In *2022 International Conference on 3D Vision (3DV)*, pages 393–402. IEEE, 2022.

[33] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. pages 3302–3312, 2019.

[34] M. Ding, Z. Wang, J. Sun, J. Shi, and P. Luo. Camnet: Coarse-to-fine retrieval for camera re-localization. pages 2871–2880, 2019.

[35] Q. Yan, J. Zheng, S. Reding, S. Li, and I. Doytchinov. Crossloc: Scalable aerial localization assisted by multimodal synthetic data. pages 17358–17368, 2022.

[36] B. Wang, C. Chen, C. X. Lu, P. Zhao, N. Trigoni, and A. Markham. Atloc: Attention guided camera localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10393–10401, 2020.

[37] F. Xue, X. Wu, S. Cai, and J. Wang. Learning multi-view camera relocalization with graph neural networks. pages 11372–11381. IEEE, 2020.

[38] A. Moreau, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *Conference on Robot Learning*, pages 1347–1356. PMLR, 2022.

[39] Y. Shavit, R. Ferens, and Y. Keller. Learning multi-scene absolute pose regression with transformers. pages 2733–2742, 2021.

[40] Y. Shavit and Y. Keller. Camera pose auto-encoders for improving pose regression. pages 140–157. Springer, 2022.

[41] S. Chen, X. Li, Z. Wang, and V. A. Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. pages 1–17. Springer, 2022.

[42] N. Roy, W. Burgard, D. Fox, and S. Thrun. Coastal navigation-mobile robot navigation with uncertainty in dynamic environments. volume 1, pages 35–40. IEEE, 1999.

[43] G. Costante, C. Forster, J. Delmerico, P. Valigi, and D. Scaramuzza. Perception-aware path planning. *arXiv preprint arXiv:1605.04151*, 2016.

[44] Z. Zhang and D. Scaramuzza. Perception-aware receding horizon navigation for mavs. pages 2534–2541. IEEE, 2018.

[45] A. Kim and R. M. Eustice. Perception-driven navigation: Active visual slam for robotic area coverage. pages 3196–3203, 2013. doi:10.1109/ICRA.2013.6631022.

[46] D. Fontanelli, P. Salaris, F. A. Belo, and A. Bicchi. Visual appearance mapping for optimal vision based servoing. In *Experimental Robotics: The Eleventh International Symposium*, pages 353–362. Springer, 2009.

[47] C. Papachristos, S. Khattak, and K. Alexis. Uncertainty-aware receding horizon exploration and mapping using aerial robots. pages 4568–4575, 2017. doi:10.1109/ICRA.2017.7989531.

[48] J. Lim, N. Lawrance, F. Achermann, T. Stastny, R. Bähnemann, and R. Siegwart. Fisher information based active planning for aerial photogrammetry. pages 1249–1255, 2023. doi:10.1109/ICRA48891.2023.10161136.

[49] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. pages 2938–2946, 2015.

[50] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. pages 6856–6864, 2017.

[51] D. S. Chaplot, E. Parisotto, and R. Salakhutdinov. Active neural localization. *arXiv preprint arXiv:1801.08214*, 2018.

[52] Q. Fang, Y. Yin, Q. Fan, F. Xia, S. Dong, S. Wang, J. Wang, L. J. Guibas, and B. Chen. Towards accurate active camera localization. pages 122–139. Springer, 2022.

[53] M. Lodel, B. Brito, A. Serra-Gómez, L. Ferranti, R. Babuska, and J. Alonso-Mora. Where to look next: Learning viewpoint recommendations for informative trajectory planning. pages 4466–4472. IEEE, 2022.

[54] L. Bartolomei, L. Teixeira, and M. Chli. Semantic-aware Active Perception for UAVs using Deep Reinforcement Learning. pages 3101–3108, 2021. ISSN 21530866. doi:10.1109/IROS51168.2021.9635893.

[55] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description, 2018. URL `https://arxiv.org/abs/1712.07629`.

[56] T. Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

[57] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021.

[58] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. doi:10.1109/CVPR.2016.445.

[59] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[60] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. pages 8601–8610, 2018.

[61] I. A. Şucan, M. Moll, and L. E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. doi:10.1109/MRA.2012.2205651. `https://ompl.kavrakilab.org`.

A Additional Pipeline Details

A.1 Data Generation Pipeline

Our training data generation pipeline is shown in Figure 4. We use HM3D single-floor scenes as input and first perform SfM reconstruction with SuperPoint features and the hloc framework to obtain a metric map consisting of 3D landmarks and camera poses. Next, we sample training waypoints throughout each scene without assuming a grid layout. At each waypoint, images are captured by sweeping the camera across multiple yaw and pitch angles. Finally, these captured images are localized against the reconstructed metric map using COLMAP, and the resulting errors provide ground-truth labels for training ActLoc. This pipeline enables large-scale automatic data generation, ensuring diverse coverage of viewpoints and scene layouts.

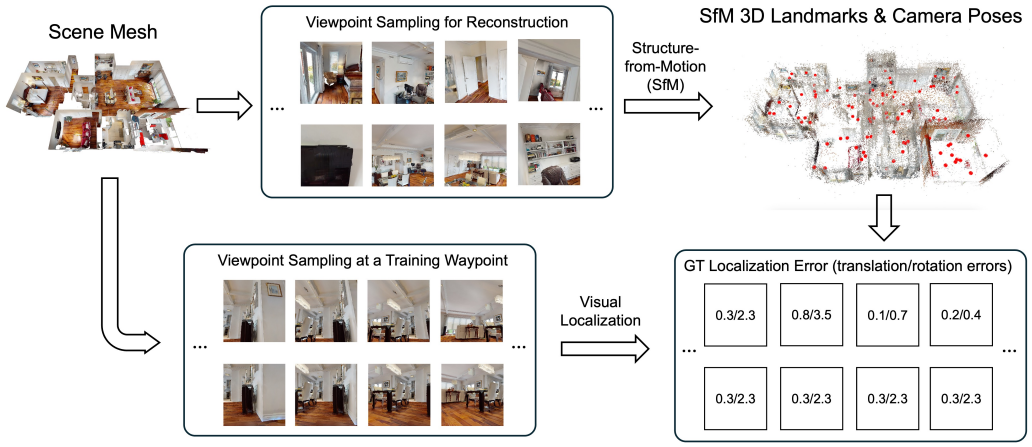

Figure 4: **Data Generation.** The pipeline first performs SfM reconstruction on HM3D scenes to obtain 3D landmarks and camera poses. Training waypoints are then sampled throughout each scene, and images captured at multiple yaw–pitch angles are localized against the reconstructed map to produce ground-truth performance labels. This process yields large-scale viewpoint-labeled data for training ActLoc.

A.2 Data Preprocessing

The data preprocessing pipeline is demonstrated in Figure 5. This pipeline first removes highly uncertain points from the 3D landmarks and implicitly encodes the training waypoint coordinate into the data via egocentric transformation. Finally, it applies a bounding box cropping on the point cloud at the input waypoint to enforce the model to focus on nearby information, and also works as a normalization.

B Additional Experiments

B.1 Robustness to Sparse Reconstruction

We evaluate the robustness of ActLoc under sparse reconstruction by progressively removing a percentage of camera poses and their associated 3D landmarks. We compare ActLoc against the baseline LWL on 10 HM3D test scenes. As shown in Table 4, ActLoc maintains slightly higher localization success rates across most sparsification levels.

B.2 Extension to Multi-class Prediction

Our model is designed with flexibility to adapt to different levels of prediction granularity. We implement a multi-class variant, ActLoc-Multi, where each viewpoint is assigned to one of four

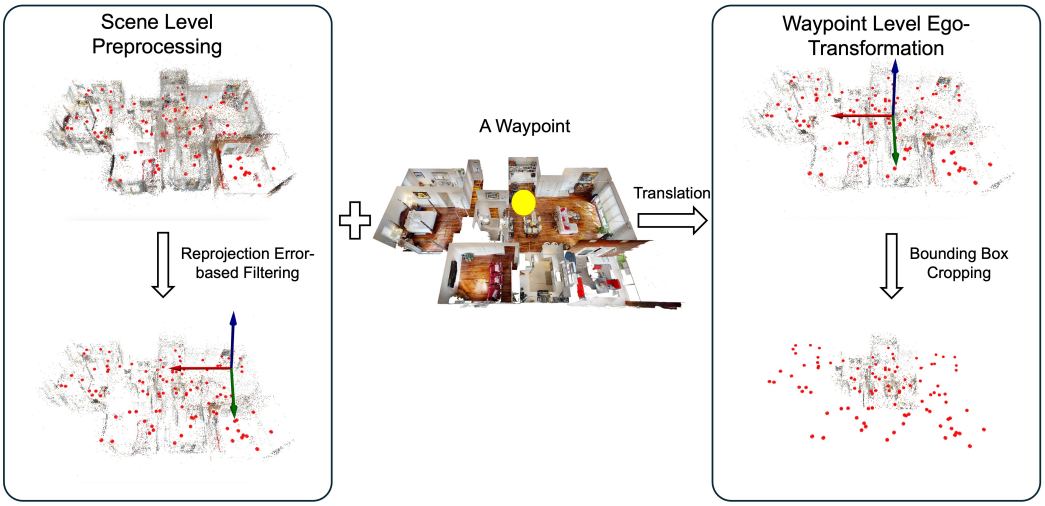

Figure 5: **Data Preprocessing.** The pipeline removes highly uncertain 3D landmarks, applies an egocentric transformation to encode the waypoint position, and crops the point cloud within a bounding box centered at the waypoint. These steps enforce input consistency, focus the model on nearby information, and normalize the scale for training.

Sparsification Level	Method	0.1 m/1°	0.25 m/2°	0.5 m/5°	5 m/10°
0%	LWL	90.04	90.98	91.17	91.92
	ActLoc	92.11	92.48	92.86	93.23
10%	LWL	89.47	90.23	90.41	91.54
	ActLoc	91.17	91.73	92.11	92.29
20%	LWL	87.78	88.72	89.66	90.04
	ActLoc	90.79	91.54	92.29	92.67
30%	LWL	88.53	89.66	90.23	90.79
	ActLoc	89.66	90.23	90.79	91.17
40%	LWL	88.72	89.66	90.60	91.73
	ActLoc	90.23	90.98	91.17	91.54
50%	LWL	89.85	90.41	90.98	91.35
	ActLoc	89.29	90.04	90.98	91.54
60%	LWL	89.10	90.23	90.79	90.79
	ActLoc	88.35	89.66	90.04	90.41
70%	LWL	87.78	88.72	89.29	89.85
	ActLoc	87.22	88.72	89.10	89.29
80%	LWL	83.83	86.09	86.84	87.59
	ActLoc	86.84	87.59	88.16	88.91
90%	LWL	83.08	84.02	84.77	85.53
	ActLoc	83.40	85.11	86.07	86.45

Table 4: **Robustness under Sparse Reconstruction.** Localization success rates (%) under different sparsification levels, measured as the percentage of camera poses and associated points removed, on ten HM3D scenes. ActLoc performs slightly better than LWL across most sparsification levels.

discrete quality levels instead of a binary label. All other training settings were kept the same as the binary model. As reported in Table 5, this extension does not improve single-viewpoint selection performance. This aligns with our discussion in the main paper: simply increasing prediction granularity is not sufficient, and further system design, together with larger-scale datasets, may be needed to make such extensions effective.

	0.1m/1°	0.25m/2°	0.5m/5°	5m/10°
ActLoc	92.11	92.48	92.86	93.23
ActLoc-Multi	88.35	89.47	89.66	90.41
Upper Bound	97.74	97.74	97.93	98.50

Table 5: **Binary vs. Multi-class ActLoc.** Comparison of binary and multi-class models on the HM3D test set, reported as localization success rates (%) at different thresholds in the single-viewpoint selection task. The binary model consistently outperforms the multi-class variant, suggesting that finer prediction granularity does not necessarily improve performance.

B.3 Experiments on Focal Axis Rotation

We evaluate the effect of in-plane (roll) rotations on visual localization using 739 viewpoints from 12 HM3D scenes. At each viewpoint, the virtual camera is rotated around its principal axis in 90° increments, and localization is performed at each angle. As summarized in Table 6, large roll angles (e.g., $\pm 90^\circ$ and 180°) result in extremely poor success rates, while the best performance is consistently observed at 0° (no in-plane rotation). These results confirm our decision to exclude in-plane rotations from the LocMap representation.

B.4 Scalability to Higher-Resolution LocMaps

We evaluate inference efficiency under different LocMap resolutions by comparing ActLoc with LWL. Both methods are tested at 6×18, 12×36, and 24×72 output grids. As shown in Table 7, ActLoc requires about 108 ms to process a single waypoint, with less than 1 ms variation across runs and resolutions. In contrast, LWL requires on average 8.3 s, 27.5 s, and 104.4 s at the three resolutions, respectively. This corresponds to a speed-up of two to three orders of magnitude, demonstrating that ActLoc scales efficiently to higher-resolution LocMaps.

B.5 Scalability to Large Scenes

We evaluate the scalability of ActLoc to large scenes by comparing its average processing time per waypoint against LWL. Scene size is measured by the number of SfM landmarks after error filtering. We divide the scenes into three ranges: $[0, 50k)$, $[50k, 100k)$, and $[100k, 150k)$. For each range, three scenes are randomly sampled, and each experiment is repeated 3 times. As shown in Table 8, the runtime of LWL increases rapidly with scene size, whereas ActLoc remains efficient, demonstrating strong scalability to larger environments.

B.6 Cross-domain Generalization with Alternative Features

We evaluate the cross-domain generalization of ActLoc by replacing the default SuperPoint features with SIFT. Experiments are conducted on five ScanNet scenes, comparing ActLoc and the LWL baseline. As shown in Table 9, ActLoc does not outperform LWL when using SIFT features. This result highlights a trade-off in our design: although processing raw inputs enables high efficiency, it can reduce robustness to alternative feature types.

B.7 Effect of LocMap Resolution

We adopt a 6×18 resolution for the LocMap as a trade-off between efficiency and accuracy. Note that ActLoc itself is trained with binary labels obtained by thresholding localization errors, while

Roll angle (deg)	0.1m/1°	0.25m/2°	0.5m/5°	5m/10°
-90	0.14	0.14	0.14	0.41
0	**84.17**	**86.20**	**87.55**	**87.96**
90	0.27	0.27	0.27	0.27
180	0.41	0.95	1.08	1.35

Table 6: **Effect of In-plane Rotations.** Localization success rates (%) under different roll angles, evaluated on 739 viewpoints from 12 HM3D scenes. Large roll angles (e.g., $\pm 90^\circ$ and 180°) lead to extremely poor localization, confirming our decision to exclude in-plane rotations from LocMap.

Method	6×18	12×36	24×72
LWL	8256 ± 56	27450 ± 92	104354 ± 545
ActLoc	$\mathbf{108.4 \pm 0.7}$	$\mathbf{108.8 \pm 0.2}$	$\mathbf{109.0 \pm 0.9}$

Table 7: **Scalability to Higher-Resolution LocMaps.** Average processing time per waypoint with standard deviation (ms). ActLoc maintains a nearly constant runtime across resolutions ($\approx$108 ms), while LWL increases from $\sim$8 s to over 100 s, resulting in a 10^2–10^3 speedup.

here we visualize the underlying continuous error maps. This provides a more detailed view of how localization difficulty varies with viewpoint resolution, from which the binary supervision is derived.

We first compare ground-truth error maps at 6×18, 12×36, and 24×72 resolutions. As shown in Figure 6, the overall error patterns remain consistent across resolutions, indicating that coarse grids already capture the dominant spatial structure. For brevity, we only show translation error; rotation error exhibits similar trends.

We further evaluate whether higher-resolution maps can be approximated via interpolation. Figure 7 compares the ground-truth 12×36 and 24×72 maps with an interpolated 24×72 map derived from the 12×36 version. The interpolated result closely matches the ground-truth 24×72 map, suggesting that ActLoc predictions at low resolution can be upsampled to higher resolutions with minimal loss.

C Qualitative Results

C.1 LocMap-based Viewpoint Prediction Visualization

We visualize examples of viewpoint prediction from our LocMap model in Figure 8. Each example shows a scene mesh with a waypoint at the center, where arrows indicate candidate viewing directions. Below, the predicted LocMap is shown as a 6×18 grid, where deep indigo cells denote viewpoints predicted as easy to localize and bright yellow-green cells as hard to localize. For the candidate viewpoints highlighted in the scene, their corresponding cells in the LocMap are connected to the actual captured images, providing an intuitive link between the prediction and scene appearance.

C.2 Path Planning Visualization

We provide additional visualizations of path planning results in two HM3D test scenes, shown in Figure 9. In each scene, we compare ActLoc against a forward-facing baseline, where planned viewpoints are shown as camera frusta in a top-down view. Regions with large differences are highlighted and further shown in zoom-in views, and for each scene the average translation and rotation errors are also reported. These examples illustrate how ActLoc selects viewpoints that improve localization accuracy while maintaining feasible paths.

Method	$< 50k$	$50k$–$100k$	$100k$–$150k$
LWL	7249	13803	25650
ActLoc	**93**	**125**	**191**

Table 8: **Scalability to Large Scenes.** Average processing time per waypoint (ms, rounded to the nearest integer) under different scene sizes, measured by the number of SfM landmarks after error filtering. ActLoc remains efficient, while LWL runtime grows rapidly with scene size.

	0.1m/1°	0.25m/2°	0.5m/5°	5m/10°
LWL [14]	36.36	36.36	36.36	36.36
ActLoc	22.22	22.22	22.22	22.22
Upper Bound	56.57	57.58	57.58	60.61

Table 9: **Cross-domain Generalization with SIFT.** Performance of SIFT-based visual localization across five ScanNet scenes, reported as localization success rates (%) at different thresholds. Unlike results with SuperPoint, ActLoc does not outperform LWL in this setting, highlighting reduced robustness to alternative feature types.

C.3 Global Score Heatmap from ActLoc-Multi

To illustrate the utility of the multi-class variant, we generate global localization score maps using ActLoc-Multi. We sample dense waypoints at a height of 0.5 m, run the model at each waypoint, and average the top five grid scores to obtain a scalar value. The resulting values are visualized as heatmaps in Figure 10. These maps reveal clear spatial variations: feature-rich regions (e.g., areas with more furniture) tend to have higher scores, indicating that ActLoc-Multi can still provide useful global cues for planning.

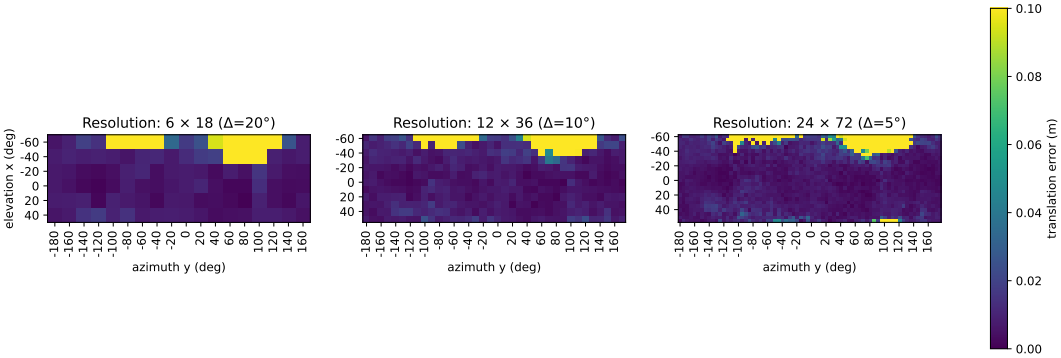

Figure 6: **Consistency Across Resolutions.** Localization error maps at resolutions of 6×18, 12×36, and 24×72 (left to right). Overall error patterns remain consistent across resolutions, showing that coarse maps already capture the main spatial structure. Translation errors are capped at 0.1 m for visualization.

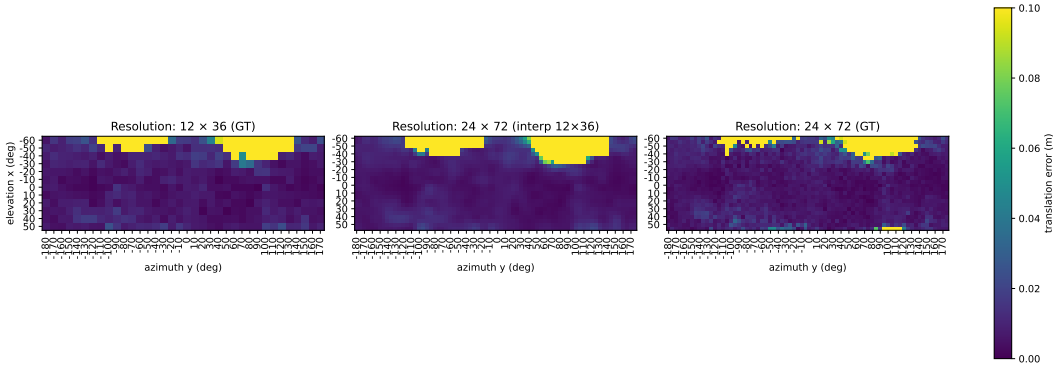

Figure 7: **Interpolation to Higher Resolutions.** Error maps from ground-truth 12×36, interpolated 24×72, and ground-truth 24×72 (left to right). Interpolated maps closely resemble the true high-resolution results, supporting upsampling as a viable strategy.

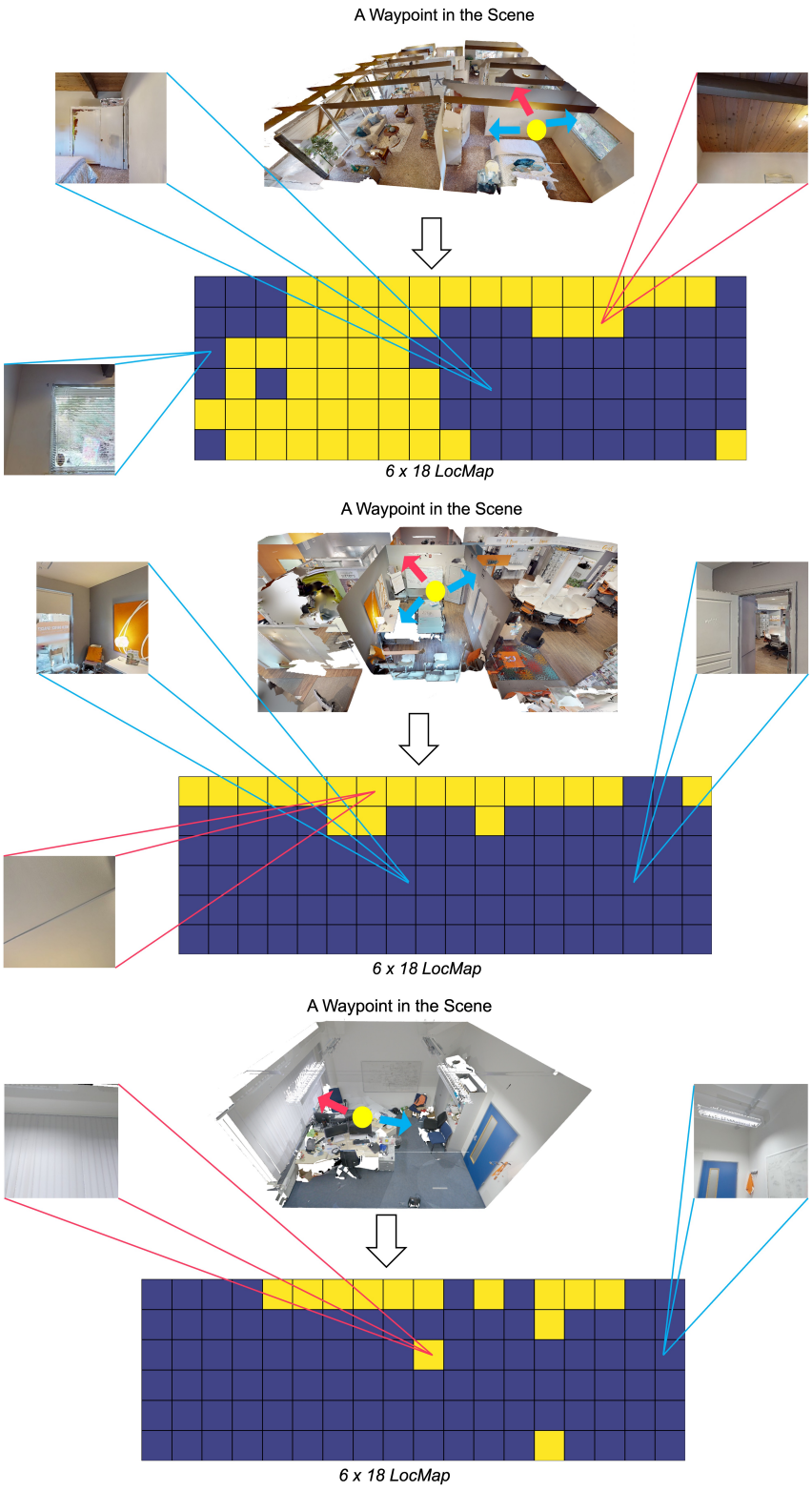

Figure 8: **Visualization of Viewpoint Prediction.** Examples from HM3D and ScanNet. Each figure shows a scene mesh with a waypoint and candidate viewing directions (arrows), the corresponding 6×18 LocMap predictions (deep indigo = easy to localize, bright yellow-green = hard to localize), and sample images for selected viewpoints.

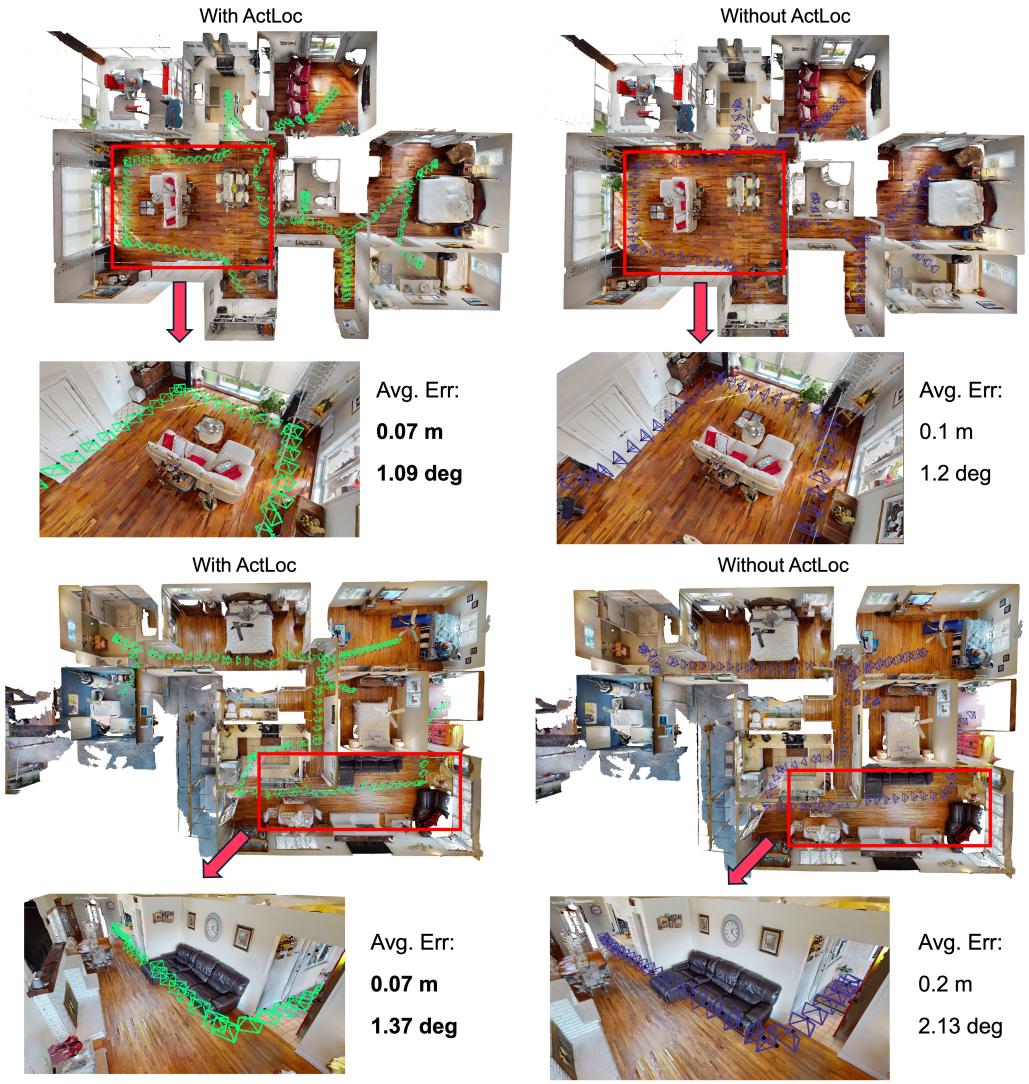

Figure 9: **Path Planning Visualization.** Comparison between ActLoc (left) and a forward-facing baseline (right) in two HM3D test scenes (top and bottom). Each viewpoint along the path is shown as a camera frustum in a top-down view. Regions with clear differences are highlighted with red boxes and further shown in zoom-in views. For each scene, the average translation and rotation errors are also reported, where ActLoc consistently achieves lower errors than the baseline.

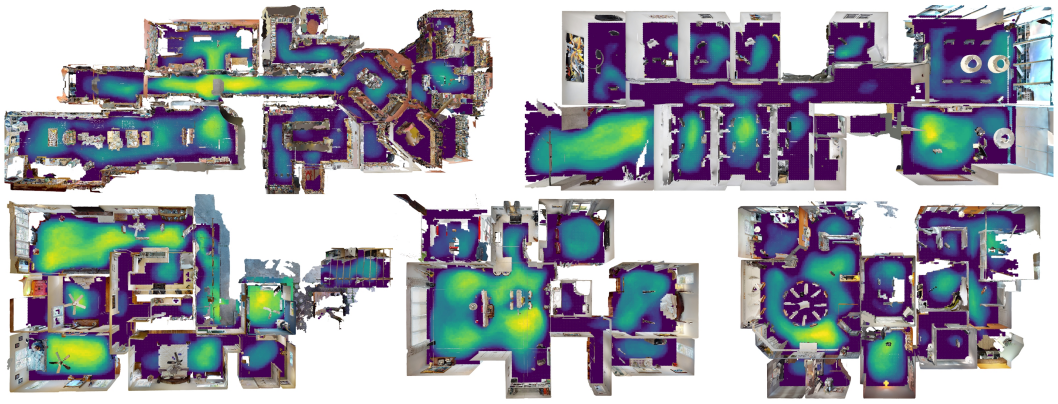

Figure 10: **Global Localization Score Maps.** Heatmap visualizations generated by ActLoc-Multi, showing averaged scores across dense waypoints in different indoor scenes. Feature-rich regions, such as areas with more furniture, exhibit higher scores, highlighting the utility of the multi-class model for capturing spatial variations in localization difficulty.