

Article

Uncertainty-Aware Bidirectional Graph Learning for Camera Localization

Hui Cai ¹, Zhiyong Li ^{2,*}, Yan Cheng ³, Fan Yang ⁴, Xin Li ⁴ and Qiang Zhai ²¹ School of Languages and Communication Studies, Chongqing University of Technology, Chongqing 400054, China; caihui@cqut.edu.cn² College of Information Engineering, Sichuan Agricultural University, Ya'an 625014, China; zhaiq@sicau.edu.cn³ PipeChina Northwest Branch, Xi'an 710016, China; chengyan02@pipechina.com.cn⁴ AIQ, Abu Dhabi 00000, United Arab Emirates; yang.fan@aiq.ae (F.Y.); li.xin@aiq.ae (X.L.)

* Correspondence: lzy@sicau.edu.cn

Abstract

Camera localization aims to estimate the six-degree-of-freedom camera pose from a single RGB image and is essential for robotics, autonomous navigation, and augmented reality. Despite recent progress in learning-based localization, robust pose estimation remains challenging under large viewpoint changes, occlusions, repetitive textures, and complex scene layouts. Existing methods mainly rely on local feature matching or independent feature regression, while the structural dependencies among multi-scale visual representations are not fully explored. To address this issue, we propose Uncertainty-aware Bidirectional Graph Learning (UBGL) for RGB-based camera localization. UBGL extracts hierarchical visual features and converts them into graph-structured representations, where nodes describe scene information at different semantic levels. A bidirectional graph learning module is introduced to exchange information between low-level geometric cues and high-level contextual features, enabling joint modeling of local correspondences and global scene structure. In addition, an uncertainty-aware relation modeling strategy estimates the reliability of graph connections and helps reduce the influence of unstable feature interactions during graph reasoning. The refined graph representations are then projected into dense scene-coordinate maps, and the final camera pose is recovered using a differentiable pose solver. Experiments on the 7Scenes and Cambridge Landmarks datasets show that UBGL achieves competitive localization accuracy, especially improving rotation estimation on the 7Scenes benchmark while maintaining comparable performance on Cambridge Landmarks. Ablation studies further demonstrate the effectiveness of bidirectional graph interaction and uncertainty-aware relation modeling.

Keywords: camera localization; Bayesian learning; bidirectional learning; graph convolutional network; multi-scale feature extraction



Academic Editor: John A. Kalomiros

Received: 17 June 2026

Revised: 9 July 2026

Accepted: 10 July 2026

Published: 15 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Camera localization refers to the estimation of the six-degree-of-freedom pose of a camera from visual observations. In particular, single-image RGB-based localization has attracted considerable attention because of its importance in many computer vision applications [1–7], such as intelligent driving [8,9], mobile robotics [10,11] and AR/VR systems [12–17]. By providing the geometric pose of the camera, localization enables mobile platforms to understand their position in the scene and perform downstream tasks such as

navigation, mapping, and path planning. However, robust localization remains challenging in unconstrained environments, where dynamic obstacles, partial occlusions, appearance variations, repeated structures, and viewpoint changes can significantly degrade pose estimation accuracy.

With the rapid development of deep neural networks, learning-based camera localization has been widely investigated in recent years. Beyond camera localization, deep learning has also significantly advanced many low-level and high-level vision tasks. In particular, transformer-based architectures have shown strong capability in modeling long-range dependencies and adaptive contextual information. Recent transformer-based models have also been applied to low-level vision tasks such as remote sensing image super-resolution. For example, TTST [18] introduces a top- k token selective transformer to reduce redundant token interactions and aggregate informative visual context. These advances motivate the exploration of structured feature interaction mechanisms for robust visual localization. Early studies formulated camera pose estimation as a direct regression problem, where a neural network predicts the camera pose from an input image. PoseNet [19] is one of the first end-to-end CNN-based frameworks for absolute camera pose regression, and a series of follow-up methods [3,20–24] have further improved this paradigm by introducing geometric constraints, temporal modeling, attention mechanisms, and prior knowledge. Although these methods are simple and efficient, their localization accuracy is still limited because directly regressing the pose vector does not explicitly model the underlying scene geometry. In particular, the translation and rotation components are highly coupled, and the rotation parameters cannot be treated as fully independent variables due to the influence of rotation order and geometric consistency.

To better exploit 3D scene structure, another line of work converts camera localization into a scene coordinate regression problem, where the network predicts dense correspondences between image pixels and 3D scene points. SANet [25] learns a scene-agnostic representation by extracting scene information from reference images and associated 3D point clouds. Zhou et al. [2] incorporated Kalman filtering into the localization framework to exploit temporal priors and likelihood estimation. DSM [26] introduced dense scene matching for scene-agnostic localization and showed its potential extension to sequential settings. Recent studies have further advanced scene coordinate regression from different perspectives, including hierarchical coordinate classification and regression [27], scene-agnostic coordinate regression [28], robust large-scale localization [29], and large-scale SCR training strategies [30]. However, many structure-based approaches still rely on non-differentiable post-processing steps, such as PnP, to obtain the final camera pose, which prevents fully end-to-end optimization, as illustrated in Figure 1. DSAC [4] and its extensions [1,31] addressed this issue by introducing differentiable pose solvers based on RANSAC and PnP. Nevertheless, these methods mainly focus on coordinate prediction and pose optimization, while the contextual dependencies among visual features are not sufficiently explored for further improving localization robustness and accuracy. Existing graph-based localization methods mainly exploit view-level or single-level feature relations, while bidirectional interactions among multi-scale visual representations [32–36] remain insufficiently modeled. In addition, most existing localization networks rely on deterministic feature affinities, which can be unreliable under occlusions, repetitive textures, weakly textured regions, or large viewpoint changes. Therefore, it remains important to develop a localization framework that can jointly capture cross-scale structural dependencies and model the reliability of graph relations.

To address these limitations, we further leverage semantic clustering techniques and graph convolutional networks for structured feature reasoning. Based on this motivation, we propose the Uncertainty-aware Bidirectional Graph Learning (dubbed UBGL) frame-

work for camera localization by modeling affinities among multi-scale deep features. First, we build multi-scale graphs to summarize and distill the semantic knowledge of CNN features and further devise a novel bidirectional communication approach to interweave graph convolution between graphs at different scale layers. Then, we adopt the Bayesian probabilistic learning [37–39] to construct a more reliable graph structure after the communication. Finally, we fuse the features re-projected from multiple layers to predict the effective scene representation, i.e., scene coordinate map. The final camera pose is estimated using a differentiable pose solver [31], which also allows gradients to be propagated back to the predicted scene coordinate map.

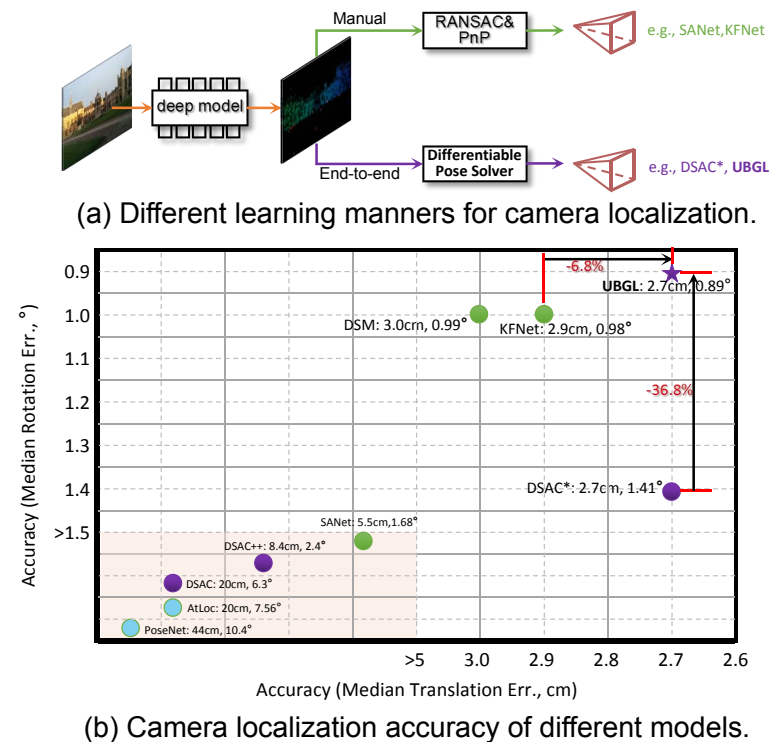


Figure 1. Learning manner and localization accuracy. (a) Prior scene coordinate regression methods require a separate PnP step, while differentiable pose solvers allow end-to-end pose learning. (b) On 7Scenes [40], UBGL achieves competitive translation accuracy and reduces the average rotation error by 36.8% compared with DSAC* [1].

Extensive experiments demonstrate that UBGL achieves competitive performance on the indoor 7Scenes dataset [40] and the outdoor Cambridge Landmarks benchmark [19]. During inference, UBGL only takes a single RGB image as input and does not require depth maps or explicit 3D scene models. During training, ground-truth camera poses and camera intrinsics are used to supervise scene-coordinate learning through reprojection consistency and subsequent differentiable pose optimization.

- **We propose a novel graph-based technique to capture scene-structure knowledge in a unified end-to-end framework.** Different from existing localization methods, our approach explores bidirectional communication among multi-scale graphs to model structural relationships for camera localization.
- **We introduce an uncertainty-aware strategy for modeling intra-graph affinities.** Instead of directly using self-correlation affinities in conventional graph convolution, the proposed method incorporates a probabilistic learning scheme to evaluate the reliability of graph relations and enhance the graph representations after bidirectional communication.

- **Competitive performance on widely used benchmarks.** Extensive experiments show that UBGL achieves competitive results on representative indoor and outdoor camera localization benchmarks, including 7Scenes [40] and Cambridge Landmarks [19]. In particular, UBGL obtains strong rotation accuracy on 7Scenes and comparable outdoor localization performance on Cambridge Landmarks.

2. Related Work

This section reviews the research most relevant to our work. For a broader and more systematic discussion of camera localization, visual SLAM, and related visual positioning techniques, readers may refer to existing survey papers [41–45].

2.1. Learning-Based Camera Localization

Learning-based camera localization methods can be broadly divided into three categories: retrieval-based relative localization [3,46–48], regression-based absolute pose localization [19,21], and structure-based absolute localization [1,2,25]. Retrieval-based methods usually rely on image retrieval techniques [49–52] to search for reference images that are visually close to the query image, and then estimate the relative pose through a coarse-to-fine localization process. For example, CAMNet [3] combines image retrieval and relative pose estimation in a cascaded framework to infer the final camera pose. These methods are generally flexible and can be applied to unseen scenes. However, their performance often depends on the quality and scale of the reference database, which may lead to high computational and storage costs during retrieval. Another representative line of work formulates camera localization as a direct pose regression problem. PoseNet [19] is one of the earliest CNN-based frameworks that directly regresses the absolute camera pose from a single image. Subsequent studies [20,48,53] followed this paradigm and improved it by introducing uncertainty modeling, geometric constraints, or additional supervision. AtLoc [21] further incorporated self-attention into the pose regression framework to capture more informative visual representations. Although direct pose regression methods are simple and efficient, their accuracy is usually limited because they do not explicitly exploit 3D scene structure or geometric correspondences. According to multi-view geometry [10,17], camera poses are strongly constrained by scene geometry. Therefore, structure-based methods aim to improve localization accuracy by recovering or exploiting scene structural information. KFNet [2] introduces a Bayesian framework to integrate temporal information from consecutive frames for scene structure learning. SANet [25] combines image retrieval and 3D point clouds to learn a scene-agnostic localization model. Recent scene coordinate regression methods have further advanced this direction. HSCNet++ [27] performs hierarchical scene coordinate classification and regression with transformer-based representations. SACReg [28] explores scene-agnostic coordinate regression to improve generalization across scenes. R-SCoRe [29] revisits scene coordinate regression for robust large-scale visual localization, while CoLoR [30] further investigates effective training strategies for large-scale scene coordinate regression. In parallel, vision foundation models have recently shown strong transferability for dense prediction, matching, and geometric perception tasks. Models such as SAM [54] and DINOv2 [55] provide general-purpose visual representations that can benefit downstream visual understanding. Although these foundation models are not specifically designed for camera localization, they highlight the importance of robust and transferable visual representations for localization-related tasks.

In general, structure-based methods can achieve strong localization performance by introducing geometric constraints. However, many early approaches still require a separate PnP step to recover the final camera pose, which limits end-to-end optimization. DSAC [4] addressed this issue by introducing a differentiable RANSAC-based pose solver, making

end-to-end camera pose learning feasible. DSAC* [1] further integrates differentiable pose optimization with deep scene coordinate regression. Nevertheless, most existing methods mainly focus on coordinate prediction and pose solving, while the contextual relationships among multi-level visual features are still not sufficiently modeled. In contrast, our UBGL explicitly explores graph-based structural reasoning and uncertainty-aware relation modeling to enhance camera localization. To further clarify the novelty of UBGL, Table 1 summarizes what UBGL adds compared with representative recent localization methods. In brief, UBGL explicitly combines multi-scale graph projection, bidirectional graph communication, uncertainty-aware relation modeling, and graph reprojection for scene-coordinate-based camera localization.

Table 1. Summary of methodological novelty. UBGL complements recent localization methods by explicitly modeling multi-scale graph communication and uncertainty-aware graph relations before scene-coordinate prediction.

Method	Main Focus	What UBGL Adds
CNN+GNN [56]	Graph reasoning for camera relocalization.	Multi-scale single-image graph construction and bidirectional cross-scale communication.
NeuMap [57]	Neural coordinate mapping.	Explicit graph-structured feature interaction before coordinate prediction.
HSCNet++ [27]	Transformer-based hierarchical coordinate regression.	Uncertainty-aware graph relation modeling rather than deterministic attention only.
CoLoR [30]	Large-scale SCR training strategy.	Feature-level graph reasoning and probabilistic relation learning.
UBGL (ours)	Uncertainty-aware bidirectional graph learning.	Jointly combines multi-scale graph projection, bidirectional graph communication, UGC, and graph reprojection.

2.2. Graph Convolutional Network

Recently, graph convolutional networks (GCNs) have shown strong capability in modeling non-Euclidean data and have been successfully applied to many vision and learning tasks, such as object detection and segmentation [58–62], crowd counting [61,63], face recognition [64], knowledge graph reasoning [65–67], 3D model retrieval [68], optical flow [69] and video/image-text retrieval [70–72]. By representing features as graph nodes and propagating information through graph edges, GCNs provide an effective way to capture long-range dependencies and structural relationships among visual representations. For example, Luo et al. [60] designed a cascaded graph model to integrate multi-scale and cross-modality cues for salient object detection. Zhang et al. [64] introduced an adaptive graph convolutional model with attention-based graph clustering for co-saliency detection. Zhai et al. [58] developed a graph learning framework to exploit the mutual benefits between camouflaged object detection and edge extraction. Nie et al. [68] proposed a multi-branch graph convolutional network for 2D-image-based 3D model retrieval, while Wang et al. [70] employed a coarse-to-fine graph neural network for video-text retrieval. More recently, graph learning has also been combined with state-space models for efficient long-range dependency modeling. For example, GraphMamba [73] adopts a graph-driven selective state space model for whole-slide image classification, showing that

graph-structured reasoning can also benefit large-scale medical image understanding tasks. These recent advances further demonstrate the flexibility of graph-based modeling beyond conventional GCN architectures.

For camera localization, Xue et al. [56] constructed a view graph to model relationships among multiple views. However, existing graph-based localization methods mainly focus on single-level graph reasoning, and the interactions among graphs built from different feature scales are still insufficiently explored. This limits their ability to jointly capture fine-grained geometric cues and high-level contextual information. To address this issue, we propose a pyramid bidirectional graph module that enables complementary information exchange across multi-scale graphs. In this way, the proposed UBGL can learn more discriminative visual representations and improve the accuracy and robustness of camera pose estimation.

2.3. Bayesian Deep Learning

Bayesian neural networks (BNNs) [38,74–78] extend standard neural networks with posterior inference and are effective for modeling predictive uncertainty. Such uncertainty-aware techniques have been successfully applied to many computer vision tasks, such as camouflaged object detection [59,79], person re-identification [80], semantic segmentation and scene parsing [37,81], and medical image analysis [82]. Yang et al. [59] further demonstrated the effectiveness of uncertainty-guided transformer reasoning for camouflaged object detection. Motivated by these works, we introduce an uncertainty-aware graph convolutional module to model intra-graph affinities from a probabilistic perspective. This design allows UBGL to evaluate the reliability of graph relations and reduces the influence of unstable feature interactions, thereby improving the stability and accuracy of camera localization.

3. Our Approach

3.1. Camera Localization Formulation

We denote the proposed camera localization network as $\mathcal{S}_{\Theta}$, where Θ represents the learnable parameters. Given a single RGB image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, the network predicts its six-degree-of-freedom (6DoF) camera pose $\mathbf{P} = [\omega, x, y, z \mid t_1, t_2, t_3]$, where $[\omega, x, y, z]$ and $[t_1, t_2, t_3]$ indicate the camera orientation and translation with respect to the world coordinate system, respectively. Given a calibrated training set $\{\mathbf{I}_i, \mathbf{K}, \mathbf{P}_i\}_{i=1}^N$, where $\mathbf{I}_i$ is the i th training image, $\mathbf{K}$ denotes the camera intrinsic matrix, and $\mathbf{P}_i$ is the corresponding ground-truth camera pose, our objective is to optimize Θ so that the model can learn discriminative visual representations and structural scene cues for accurate camera localization.

3.2. Overview

This work introduces an Uncertainty-aware Bidirectional Graph Learning framework, namely **UBGL**. As illustrated in Figure 2, the proposed framework is organized into three main modules:

- **Multi-scale Graph Projection** (Section 3.3.1). Given an image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, our multi-scale graph projection, denoted as f_{MGP} , contains two parts: multi-scale feature extractor (backbone) f_{MFE} and graph projection module f_{GProj} . f_{MFE} takes the image $\mathbf{I}$ as input, and produces feature embeddings $\mathbf{F} \in \mathbb{R}^{h \times w \times c}$. Then, $\mathbf{F}$ are encoded into a hierarchical form $\{\mathbf{F}_l\}_{l=1}^L$ by pyramid pooling: $\mathbf{I} \xrightarrow{f_{\text{MFE}}} \{\mathbf{F}_l\}_{l=1}^L$. Following [58], f_{GProj} projects $\{\mathbf{F}_l\}_{l=1}^L$ to the graph domain to generate multi-scale graph representations: $\mathcal{G} = \{\mathcal{G}_l = (\mathcal{V}_l, \mathcal{E}_l)\}_{l=1}^L: \{\mathbf{F}_l\}_{l=1}^L \xrightarrow{f_{\text{GProj}}} \{\mathcal{G}_l\}_{l=1}^L$.

- **Bidirectional Graph Communication** (Section 3.3.2). The hierarchical graphs $\mathcal{G}_l$ are further transferred by level-independent GCNs to generate the residual graphs $\mathcal{G}_l^1$ and $\mathcal{G}_l^2$. Then, the bidirectional interaction flows f_{fwc} and f_{bwc} are carefully designed for information communication through Cross Attention Transform (CAT) f_{CAT} , which is used to transfer valuable knowledge between different levels. Formally, $f_{\text{fwc}}: \{\mathcal{G}_l^1\} \xrightarrow{f_{\text{CAT}}} \{\mathcal{G}_l^f\}$ and $f_{\text{bwc}}: \{\mathcal{G}_l^2\} \xrightarrow{f_{\text{CAT}}} \{\mathcal{G}_l^b\}$. Last, $\{\mathcal{G}_l^f\}$ and $\{\mathcal{G}_l^b\}$ are summed together to yield the enhanced graphs $\{\mathcal{G}_l\}$.
- **Uncertainty-aware Graph Convolution** (Section 3.3.3). Finally, the Bayesian-learning-based uncertainty-aware graph convolution f_{UGC} takes $\{\mathcal{G}_l\}$ as input and produces the learnable graph adjacency matrix to refine the graph structure $\{\mathcal{G}_l^*\}: \{\mathcal{G}_l\} \xrightarrow{f_{\text{UGC}}} \{\mathcal{G}_l^*\}$. Next, the graph reprojection operation f_{Reproj} is used to transform the refined pyramid graphs back into the image-coordinate space, yielding the concatenated feature representation $\mathbf{F}^* = \text{cat}(\{\mathbf{F}_l^*\}_{l=1}^L) = \text{cat}(f_{\text{Reproj}}(\{\mathcal{G}_l^*\}_{l=1}^L))$. The scene coordinate map SC is then predicted from $\mathbf{F}^*$ through f_{FC} , i.e., $\mathbf{F}^* \xrightarrow{f_{\text{FC}}} SC$. Finally, the estimated scene coordinate map is passed to a differentiable pose solver [1] to recover the final camera pose.

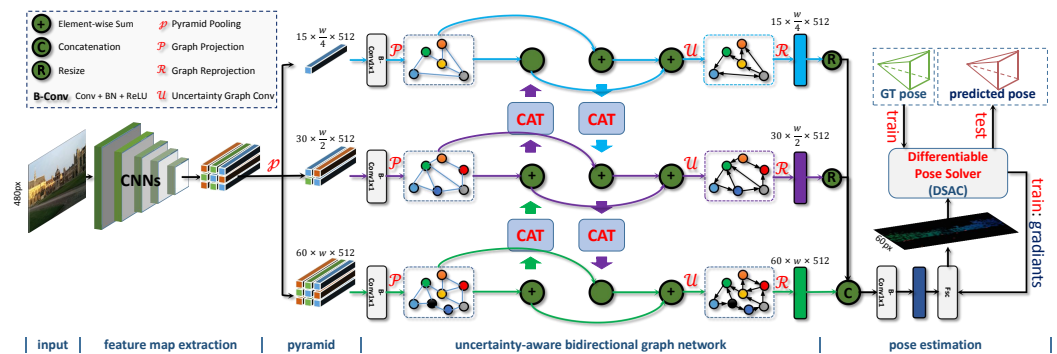


Figure 2. Architecture of the proposed UBGL framework for camera localization. The model constructs multi-scale graph representations from CNN features, performs bidirectional cross-graph interaction through the cross-graph attention transform module, and predicts the scene coordinate map for final pose estimation. More implementation details are provided in Section 3.3.

3.3. Uncertainty-Aware Bidirectional Graph Learning

Before introducing each module, we first define the dimensional convention used in the following sections. For the l th feature level, the feature map is denoted as $\mathbf{F}_l \in \mathbb{R}^{H_l \times W_l \times C}$. We flatten it into $\mathbf{X}_l \in \mathbb{R}^{S_l \times C}$, where $S_l = H_l W_l$. The graph projection module maps $\mathbf{X}_l$ into K_l graph nodes, represented by $\mathbf{V}_l \in \mathbb{R}^{K_l \times C}$. The corresponding soft assignment matrix is denoted as $\mathbf{Q}_l \in \mathbb{R}^{S_l \times K_l}$, where each row records the assignment weights from one spatial position to all graph nodes. The intra-graph adjacency matrix is $\mathcal{A}_l \in \mathbb{R}^{K_l \times K_l}$. After graph reasoning and uncertainty-aware graph convolution, the refined graph representation remains $\mathbf{V}_l^* \in \mathbb{R}^{K_l \times C}$. Finally, graph reprojection maps the refined graph nodes back to the image domain by $\mathbf{X}_l^* = \mathbf{Q}_l \mathbf{V}_l^* \in \mathbb{R}^{S_l \times C}$, which is then reshaped into $\mathbf{F}_l^* \in \mathbb{R}^{H_l \times W_l \times C}$.

3.3.1. Multi-Scale Graph Projection

The multi-scale graph projection module f_{MGP} aims to transform dense image features into hierarchical graph representations, thereby enabling structured reasoning at multiple semantic scales. Given $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, we first use a CNN-based feature encoder f_{FE} , e.g., a ResNet-FCN backbone, to extract a high-level visual embedding:

$$\mathbf{F} = f_{\text{FE}}(\mathbf{I}), \quad (1)$$

where $\mathbf{F} \in \mathbb{R}^{h \times w \times c}$ is the resulting feature tensor with spatial resolution $h \times w$ and c channels. It encodes the global semantic response while retaining the original spatial arrangement. We then perform multi-scale pooling over $\mathbf{F}$ to generate a pyramid of feature maps $\{\mathbf{F}_l\}_{l=1}^L$:

$$\{\mathbf{F}_l\}_{l=1}^L = f_{\text{MFE}}(\mathbf{F}). \quad (2)$$

Finally, each feature level is mapped into a semantic graph representation $\mathcal{G}_l = (\mathcal{V}_l, \mathcal{A}_l)$ by applying the graph projection function f_{GProj} independently. Following [58,83,84], this projection is achieved through learnable local residual aggregation. For the l th level, we use learnable anchor centers $\mathbf{W}_l \in \mathbb{R}^{K_l \times C}$ and scale parameters $\sigma_l \in \mathbb{R}^{K_l \times C}$. Each vector $w_{l,j}$ in $\mathbf{W}_l$ represents the anchor center of the j th graph node.

$$v_{l,j} = \frac{v'_{l,j}}{\|v'_{l,j}\|_2}, \quad v'_{l,j} = \frac{1}{\sum_{i=1}^{S_l} q_{l,ij}} \sum_{i=1}^{S_l} q_{l,ij} \left(\frac{x_{l,i} - w_{l,j}}{\sigma_{l,j}} \right), \quad (3)$$

where $x_{l,i}$ is the i th row of $\mathbf{X}_l$, $\sigma_{l,j}$ is the j th scale vector in σ_l , and $v_{l,j}$ is the j th node feature in $\mathcal{V}_l$. The soft assignment weight $q_{l,ij}$ between spatial feature $x_{l,i}$ and anchor $w_{l,j}$ is computed as

$$q_{l,ij} = \frac{\exp\left(-\left\|\frac{x_{l,i} - w_{l,j}}{\sigma_{l,j}}\right\|_2^2 / 2\right)}{\sum_{k=1}^{K_l} \exp\left(-\left\|\frac{x_{l,i} - w_{l,k}}{\sigma_{l,k}}\right\|_2^2 / 2\right)}. \quad (4)$$

Collecting all assignment weights gives $\mathcal{Q}_l \in \mathbb{R}^{S_l \times K_l}$. Given the node matrix $\mathcal{V}_l \in \mathbb{R}^{K_l \times C}$, the adjacency matrix is constructed by normalizing pairwise node correlations:

$$\mathcal{A}_l = f_{\text{norm}}(\mathcal{V}_l \mathcal{V}_l^T) \in \mathbb{R}^{K_l \times K_l}. \quad (5)$$

where $\mathcal{A}_l$ denotes the normalized node-relation matrix at the l th feature level, and f_{norm} is the normalization function.

3.3.2. Bidirectional Graph Communication

BGC aims at explicitly establishing communication between different scales and contains three operations/functions: (1) Graph Reasoning f_{GR} , (2) Bidirectional (forward/backward) Communication $f_{\text{fWC}}, f_{\text{bWC}}$, and (3) Fusion f_{fuse} .

(1) Graph Reasoning f_{GR} . Given the multi-scale graphs produced by the graph projection module, we apply intra-graph reasoning to each graph $\mathcal{G} * l$ through $f * \text{GR}$ to refine its node representations. In our implementation, f_{GR} is instantiated with graph convolution [85]. At each scale, two graph reasoning branches are introduced to support forward and backward information propagation. These branches perform graph convolution in parallel while allowing their reasoning processes to interact, thereby enabling bidirectional communication within the graph representation:

$$\mathcal{V}_l^k = f_{\text{GR}}^k(\mathcal{V}_l, \mathcal{A}_l) = g(\mathcal{A}_l \mathcal{V}_l W_l^k), \quad k = 1, 2, \quad (6)$$

where $W_l^k \in \mathbb{R}^{C \times C}$ is the learnable transformation matrix, and $\mathcal{V}_l^k \in \mathbb{R}^{K_l \times C}$ keeps the same node dimension as $\mathcal{V}_l$.

(2) Bidirectional Communication. The knowledge transfer scheme is the key for communication. Inspired by [86], we carefully design a knowledge transfer module to effectively communicate valuable information between graphs at different scales. As shown in Figure 3, the cross-attention transform module f_{CAT} transfers information from a source graph to a target graph. Given a source graph node matrix $\mathcal{V}_s \in \mathbb{R}^{K_s \times C}$ and a target graph

node matrix $\mathcal{V}_t \in \mathbb{R}^{K_t \times C}$, CAT first computes the cross-graph attention matrix from target nodes to source nodes:

$$\mathcal{M}_{s \rightarrow t} = f_{\text{norm}}(\mathcal{V}_t \mathcal{V}_s^T) \in \mathbb{R}^{K_t \times K_s}. \quad (7)$$

The source information transferred to the target graph is then obtained by

$$\hat{\mathcal{V}}_t = \mathcal{M}_{s \rightarrow t} \mathcal{V}_s \in \mathbb{R}^{K_t \times C}. \quad (8)$$

Finally, the target graph is updated as

$$\mathcal{V}'_t = f_{\text{GR}}(\mathcal{V}_t + \chi \hat{\mathcal{V}}_t), \quad (9)$$

where χ controls the contribution of the transferred source information. This formulation keeps the output dimension identical to the target graph and therefore allows communication between graph levels with different node numbers K_s and K_t .

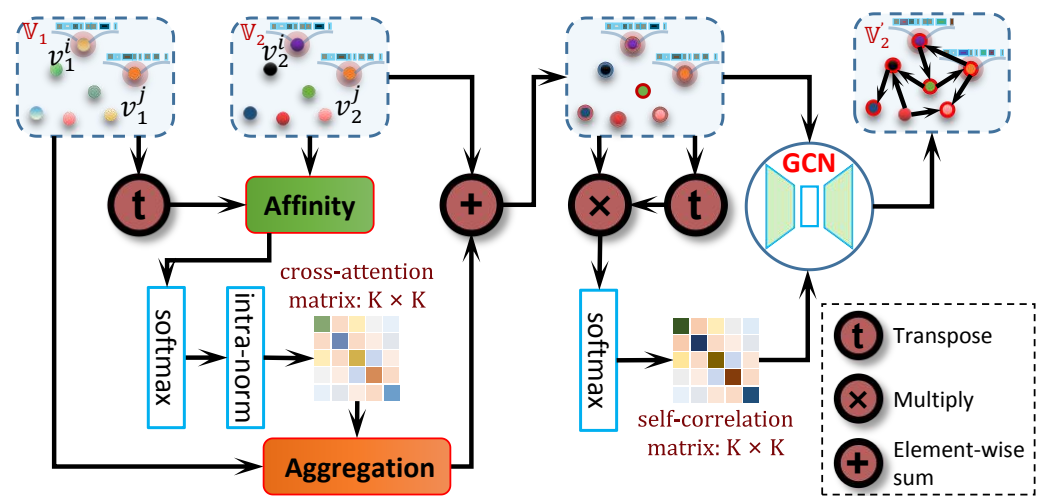


Figure 3. Illustration of the Cross-graph Attention Transform. CAT performs information exchange between two graph representations. Given two graphs $\mathbb{G}_1 = (\mathbb{V}_1, \mathbb{E}_1)$ and $\mathbb{G}_2 = (\mathbb{V}_2, \mathbb{E}_2)$, the proposed transform uses the structural and semantic relations across graphs to update $\mathbb{V}_2$ into the enhanced representation $\mathbb{V}'_2$. More details are provided in Section 3.3.2.

Based on the above CAT operation, we implement bidirectional communication across feature levels. The forward flow f_{fwc} transfers information from high-level semantic graphs to lower-level graphs:

$$\mathcal{V}_{l-1}^f = f_{\text{CAT}}(\mathcal{V}_l^1, \mathcal{V}_{l-1}^1), \quad l = 2, \dots, L, \quad (10)$$

where $\mathcal{V}_l^1$ is the source graph, and $\mathcal{V}_{l-1}^1$ is the target graph. The output $\mathcal{V}_{l-1}^f$ has the same dimension as the target graph, i.e., $\mathcal{V}_{l-1}^f \in \mathbb{R}^{K_{l-1} \times C}$. The top-level graph $\mathcal{V}_L^f$ is directly inherited from the graph reasoning branch.

In contrast, the backward flow f_{bwc} transfers information from lower-level graphs to higher-level graphs:

$$\mathcal{V}_{l+1}^b = f_{\text{CAT}}(\mathcal{V}_l^2, \mathcal{V}_{l+1}^2), \quad l = 1, \dots, L-1, \quad (11)$$

where $\mathcal{V}_l^2$ is the source graph, and $\mathcal{V}_{l+1}^2$ is the target graph. The output $\mathcal{V}_{l+1}^b$ has the same dimension as the target graph, i.e., $\mathcal{V}_{l+1}^b \in \mathbb{R}^{K_{l+1} \times C}$. The bottom-level graph $\mathcal{V}_1^b$ is directly inherited from the graph reasoning branch.

(3) *Fusion* f_{fuse} . After forward and backward communication, the enhanced graph representations are fused at each level by element-wise summation:

$$\check{\mathcal{V}}_l = \mathcal{V}_l^f + \mathcal{V}_l^b \in \mathbb{R}^{K_l \times C}. \quad (12)$$

Since both $\mathcal{V}_l^f$ and $\mathcal{V}_l^b$ are defined at the same feature level, they have identical dimensions and can be directly fused.

3.3.3. Uncertainty-Aware Graph Convolution

We formulate the uncertainty-aware graph convolution module f_{UGC} as a probabilistic representation learning process to estimate the reliability of intra-graph affinities. Different from conventional graph convolution that directly uses a deterministic affinity value, such as a self-correlation matrix, f_{UGC} predicts a probability distribution for each edge relation between two graph nodes. In this work, the edge affinity is modeled with a Gaussian distribution parameterized by the mean μ_e and the variance σ_e . Therefore, the correlation score c_e between two nodes is treated as a random variable sampled from the learned distribution:

$$c_e \sim \mathcal{N}(\mu_e, \sigma_e), \quad (13)$$

where μ_e and σ_e are generated by UGC, as illustrated in Figure 4.

For a graph at level l , each edge relation between node i and node j is represented by a random affinity variable $c_{l,ij}$. Instead of using a single deterministic correlation score, UGC predicts the mean $\mu_{l,ij}$ and standard deviation $\sigma_{l,ij}$ of this affinity distribution from the node features. The mean represents the expected relation strength, while the standard deviation reflects the uncertainty of the relation. This probabilistic formulation allows graph convolution to aggregate information over sampled relation matrices rather than relying on one fixed adjacency matrix.

Specifically, we employ the reparameterization strategy to sample edge affinities. For the t th sampling step, a noise variable $\epsilon_{l,ij}^{(t)}$ is drawn from a standard Gaussian distribution, i.e., $\epsilon_{l,ij}^{(t)} \sim \mathcal{N}(0, 1)$. The sampled affinity is then obtained as

$$c_{l,ij}^{(t)} = \mu_{l,ij} + \sigma_{l,ij} \epsilon_{l,ij}^{(t)}, \quad t = 1, \dots, T. \quad (14)$$

After drawing T samples, the sampled affinities are averaged and normalized to obtain the uncertainty-aware affinity matrix $\mathcal{A}_l^u$. Finally, $\mathcal{A}_l^u$ is used in graph convolution to further refine the pyramid graph representations:

$$\mathcal{V}_l^* = f_{\text{UGC}}(\check{\mathcal{V}}_l) = g(\mathcal{A}_l^u \check{\mathcal{V}}_l W_l) \in \mathbb{R}^{K_l \times C}. \quad (15)$$

Here, $\mathcal{A}_l^u \in \mathbb{R}^{K_l \times K_l}$ is the uncertainty-aware adjacency matrix for the l th graph level, and $W_l \in \mathbb{R}^{C \times C}$ is a learnable graph-convolution weight matrix.

To recover dense image-level features from the refined graph representations, we reuse the soft assignment weights obtained during graph projection. For the l th feature level, the assignment matrix is denoted as $\mathcal{Q}_l = [q_{l,ij}] \in \mathbb{R}^{S_l \times K_l}$, where $q_{l,ij}$ records the assignment response from the i th spatial position to the j th graph node. The graph reprojection operation f_{Rproj} is defined as

$$\mathbf{X}_l^* = f_{\text{Rproj}}(\mathcal{V}_l^*) = \mathcal{Q}_l \mathcal{V}_l^*, \quad \mathcal{Q}_l \in \mathbb{R}^{S_l \times K_l}, \quad \mathbf{X}_l^* \in \mathbb{R}^{S_l \times C}. \quad (16)$$

The reprojected matrix $\mathbf{X}_l^*$ is then reshaped into $\mathbf{F}_l^* \in \mathbb{R}^{H_l \times W_l \times C}$, which denotes the reprojected feature map at the l th scale.

The reprojected feature maps $\{\mathbf{F}_l^*\}_{l=1}^L$ from different scales are first resized to a common spatial resolution and then concatenated to form the aggregated feature representation. Based on this fused representation, the scene coordinate map $\mathcal{SC}$ is regressed. Finally, a differentiable pose solver [1] is adopted to estimate the camera pose and propagate gradients back to the scene coordinate prediction, enabling end-to-end optimization.

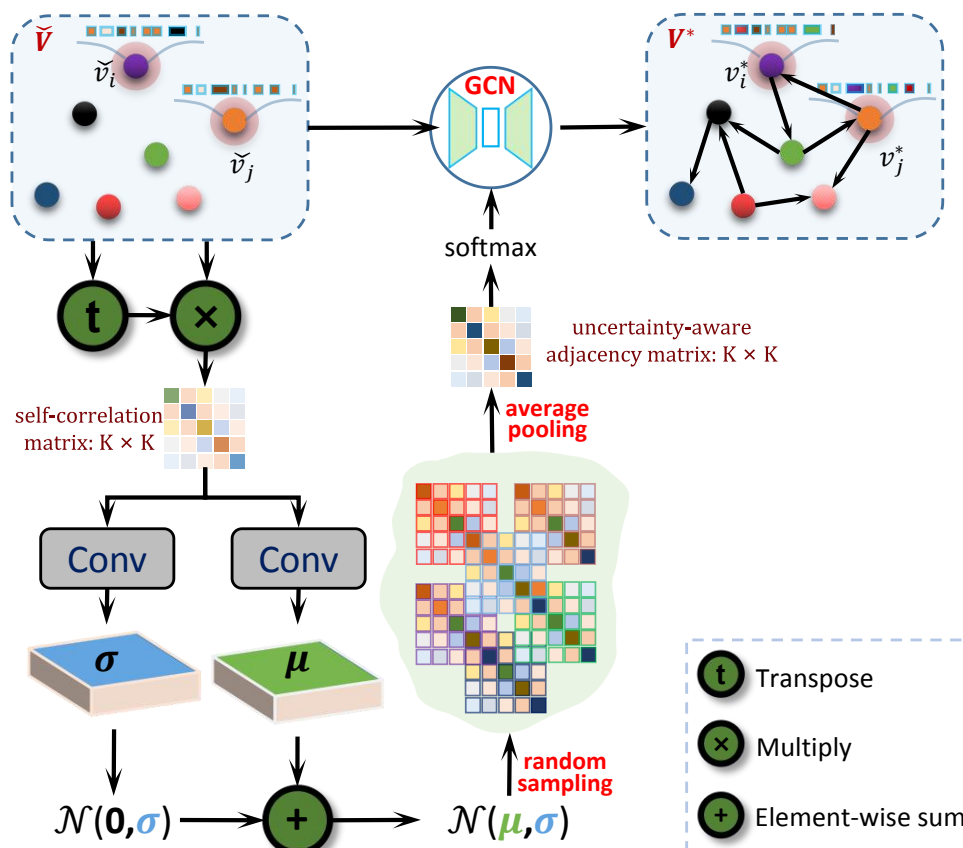


Figure 4. Illustration of UGC. UGC predicts the mean and variance of graph relations, samples probabilistic edge affinities through reparameterization, averages sampled affinities to obtain an uncertainty-aware adjacency matrix, and performs graph convolution with the estimated relation matrix. The module is used to reduce the sensitivity of graph reasoning to unstable feature relations.

3.4. Implementation Details

3.4.1. Multi-Scale Graph Projection Module

Following existing scene-coordinate regression methods [1], we employed ResNet-18 [87] as the backbone. We constructed $N = 3$ feature levels by performing average pooling on the backbone output feature map. Following [84], f_{proj} was designed based on learnable residual aggregation, where feature residuals were computed with respect to a set of learnable anchor centers. The resulting multi-scale feature maps $\{\mathbf{F}_l\}_{l=1}^N$ were then projected into semantic graphs with $K_l \in \{8, 16, 32\}$ nodes for the three feature levels, respectively.

3.4.2. Bidirectional Graph Communication Module

For CAT, Equations (7)–(9) define the cross-graph information propagation process. In our implementation, we set $\chi = 1.0$. The function f_{GR} was realized by learnable GCNs [85] with a ReLU activation layer.

3.4.3. Uncertainty-Aware Graph Learning

The mean and variance of intra-graph relations were implemented by MLPs (3×1 convolution layers), respectively. No manually tuned uncertainty threshold was used in UGC. Instead, the relation mean and variance were predicted by learnable layers and optimized jointly with the localization objective. The reparameterized sampling followed a standard Gaussian noise prior, i.e., $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The main hyperparameter related to uncertainty estimation was the Monte Carlo sampling number T , whose sensitivity is analyzed in Section 4.

3.4.4. Prediction Head and Optimization Objective

Given the refined graph features, we employed a prediction head composed of three fully connected layers to generate the scene coordinate map $\mathcal{SC}$. The entire network was optimized with a coarse-to-fine training schedule. During the first 50 epochs, the model was trained with the reprojection loss to stabilize scene coordinate prediction:

$$L_{\text{reproj}}(y, h) = \|p - \mathbf{K}h^{-1}y\|. \quad (17)$$

It should be noted that RGB images are the only network input at both training and inference stages. During training, the ground-truth camera pose h and intrinsic matrix $\mathbf{K}$ are used only to define geometric supervision for the predicted scene coordinates through reprojection consistency. The method does not require depth maps or explicit 3D scene models as network inputs, but it is not a purely self-supervised objective. Since reprojection supervision alone can be ambiguous along camera rays, we used it only as a warm-up objective to stabilize the initial scene-coordinate prediction.

In practice, directly optimizing the differentiable pose loss from random initialization can be unstable because early scene-coordinate predictions are noisy. Therefore, we adopted the above coarse-to-fine training schedule. The reprojection loss was first used for 50 epochs to warm up the scene-coordinate prediction, and the differentiable pose loss was then applied for another 50 epochs to optimize the final camera pose. We did not use additional special training tricks beyond the reported data augmentation, Adam optimizer, and the two-stage training schedule. This strategy led to stable convergence in our experiments. Specifically, the gradient of the pose loss with respect to the predicted scene coordinates was computed through differentiable sampling and pose estimation:

$$L_{\text{pose}}(y, h) = E_{j \sim p(j|y)} [l_{\text{pose}}(y, h)], \quad (18)$$

$$\frac{\partial}{\partial y} L_{\text{pose}}(y, h) = E_j \left[l_{\text{pose}}(y, h) \frac{\partial}{\partial y} \log p(j|y) + \frac{\partial}{\partial y} l_{\text{pose}}(y, h) \right]. \quad (19)$$

Here, $l_{\text{pose}}(y, h)$ denotes the pose loss of a hypothesis obtained by applying PnP to a randomly sampled set of scene coordinates from y . The term $L_{\text{pose}}(y, h)$ represents the expected pose loss over all hypotheses sampled according to the probability distribution $p(j|y)$. Unlike the single-point reprojection warm-up, this pose-level objective constrains the predicted scene coordinates through multi-point geometric consistency. In this two-stage optimization, the warm-up stage encourages each predicted scene coordinate to be geometrically compatible with its image location under the known training pose and intrinsics, while the pose-level stage evaluates whether a sampled set of predicted coordinates can jointly recover the correct camera pose. Degenerate or unreliable coordinate predictions that may satisfy isolated reprojection constraints but cannot support a consistent camera pose lead to large pose errors and are therefore discouraged during optimization. In this way, the training process first stabilizes coordinate prediction with reprojection consistency and then refines it using differentiable pose supervision.

4. Experiments

4.1. Evaluation Protocol

4.1.1. Benchmark Datasets

Following prior camera localization studies [1,2,4], we evaluated the proposed method on one indoor benchmark and one outdoor benchmark:

- The 7Scenes dataset [40] is an RGB-D camera localization benchmark composed of seven small indoor scenes. The images and ground-truth camera poses were collected using a handheld KinectFusion system [88]. Each scene contains thousands of training and testing images, covering challenging factors such as motion blur, reflective surfaces, repetitive structures, and textureless regions.
- The Cambridge Landmarks dataset [19] is a large-scale outdoor localization benchmark captured around the University of Cambridge, UK. Compared with indoor benchmarks, this dataset contains fewer images per scene, typically on the order of hundreds, but covers a much larger spatial area and more complex outdoor appearance variations.

For both datasets, UBGL took a single RGB image as the network input during both training and inference. During training, the corresponding ground-truth camera poses and camera intrinsics were used only as geometric supervision for scene-coordinate learning and differentiable pose optimization. During inference, no ground-truth camera pose, camera intrinsics for supervision, depth map, or explicit 3D scene model was required as a network input.

4.1.2. Evaluation Metric

Consistent with existing camera localization studies [1,2,4,26], we evaluated pose estimation accuracy with the median rotation error (MRE , $^{\circ}$) and median translation error (MTE , m). The scene-level median errors were further averaged across all scenes to summarize the final localization performance. For the Cambridge Landmarks benchmark, we report two averaging protocols to avoid ambiguity. Avg-4 was computed over KingsCollege, ShopFacade, StMarysChurch, and OldHospital, following the commonly reported four-scene protocol in several prior works. Avg-5 additionally includes GreatCourt when the corresponding result was available.

4.1.3. Representative Baselines

We compared UBGL with representative camera localization methods rather than claiming to include all recently published methods. The compared methods covered several major categories, including absolute pose regression, retrieval-based localization, relative pose estimation, and scene coordinate regression. We also included transformer-related localization methods when reported results were available, such as TransPoseNet [89], and discussed HSCNet++ [27] in related work. Specifically, we compared our approach against absolute pose regression networks such as PoseNet [19], PoseNet17 [20], MapNet [22], and AtLoc [21]. We also compared with retrieval or relative-pose-based methods, including CAMNet [3], InLoc [90], and RelocNet [46]. In addition, we included structure-based and scene-coordinate-regression methods such as KFNet [2], SANet [25], DSAC [4], DSAC++ [31], DSAC* [1], DSM [26], and NeuMap [57].

The results of competing methods were collected from their original publications. Since different methods may use different input modalities, reference images, temporal information, 3D point clouds, depth maps, scene-specific training strategies, or different training protocols, the comparison should be interpreted as a representative benchmark

comparison rather than a strictly controlled comparison under identical settings. Therefore, we avoid drawing overly broad conclusions from these results.

4.1.4. Training Configuration

UBGL was trained from scratch, and all learnable parameters were initialized randomly, including those in the ResNet-18 backbone [87]. For image preprocessing, we preserved the original aspect ratio and normalized the input resolution by resizing the image height to 480 pixels. During training, several data augmentation strategies were applied, including Gaussian blurring, color perturbation, and random rescaling with scale factors sampled from [0.8, 1.2]. We optimized the network using Adam, with the initial learning rate set to 1×10^{-4} .

4.1.5. Implementation Environment

The proposed UBGL framework was developed with PyTorch 1.13.1. The training stage was carried out on an NVIDIA Tesla V100 GPU to support a larger batch size. During evaluation, all models were tested on an NVIDIA GTX Titan X GPU equipped with 12 GB of memory.

4.2. Comparison with Existing Methods

4.2.1. Indoor Localization on the 7Scenes Dataset [40]

As reported in Table 2, UBGL achieves the best average rotation error and a tied best average translation error after rounding among the compared methods. Specifically, UBGL obtains an average translation error of 0.027 m, which is comparable to DSAC* [1]. The main advantage of UBGL lies in the rotation estimation: compared with KFNet [2], UBGL reduces the average MRE from 0.98° to 0.89° , and compared with DSAC* [1], it reduces the average MRE from 1.411° to 0.89° . These results indicate that UBGL can better handle visual ambiguity in indoor camera localization and produce more reliable pose estimates. The trajectory visualization in Figure 5 further demonstrates that UBGL provides more accurate localization results than the compared models.

Table 2. Evaluation on the 7Scenes indoor localization benchmark [40]. The numbers of competing methods are collected from their original publications, and the top-performing entries are marked in bold. Since different methods may use different input information, training protocols, or geometric priors, the comparison should be interpreted as a representative benchmark comparison rather than a strictly controlled comparison.

Scene Scene Scale	Chess $3 \times 2 \text{ m}^2$	Fire $2.5 \times 1 \text{ m}^2$	Heads $2 \times 0.5 \text{ m}^2$	Office $2.5 \times 2 \text{ m}^2$	Pumpkin $2.5 \times 2 \text{ m}^2$	RedKitchen $4 \times 3 \text{ m}^2$	Stairs $2.5 \times 2 \text{ m}^2$	Avg.
PoseNet15 [19]	0.32 m, 8.12°	0.47 m, 14.4°	0.29 m, 12.0°	0.48 m, 7.68°	0.47 m, 8.42°	0.59 m, 8.64°	0.47 m, 13.8°	0.44 m, 10.4°
Active Search [91]	0.04 m, 1.96°	0.03 m, 1.53°	0.02 m, 1.45°	0.09 m, 3.61°	0.08 m, 3.10°	0.07 m, 3.37°	0.03 m, 2.22°	0.051 m, 2.46°
LSTM+Pose [92]	0.24 m, 5.77°	0.34 m, 11.9°	0.21 m, 13.7°	0.30 m, 8.08°	0.33 m, 7.00°	0.37 m, 8.83°	0.40 m, 13.7°	0.31 m, 9.85°
Hourglass [93]	0.15 m, 6.17°	0.27 m, 10.84°	0.19 m, 11.63°	0.21 m, 8.48°	0.25 m, 7.01°	0.27 m, 10.15°	0.29 m, 12.46°	0.23 m, 9.53°
BranchNet [94]	0.18 m, 5.17°	0.34 m, 8.99°	0.20 m, 14.15°	0.30 m, 7.05°	0.27 m, 5.10°	0.33 m, 7.40°	0.38 m, 10.26°	0.29 m, 8.30°
VidLoc [5]	0.18 m, -	0.26 m, -	0.14 m, -	0.26 m, -	0.36 m, -	0.31 m, -	0.26 m, -	0.25 m, -
DSAC [4]	0.02 m, 1.2°	0.04 m, 1.5°	0.03 m, 2.7°	0.04 m, 1.6°	0.05 m, 2.0°	0.05 m, 2.0°	1.17 m, 33.1°	0.20 m, 6.3°
RelocNet [46]	0.12 m, 4.14°	0.26 m, 10.4°	0.14 m, 10.5°	0.18 m, 5.32°	0.26 m, 4.17°	0.23 m, 5.08°	0.28 m, 7.53°	0.21 m, 6.73°
MapNet [22]	0.08 m, 3.25°	0.27 m, 11.69°	0.18 m, 13.25°	0.17 m, 5.15°	0.22 m, 4.02°	0.23 m, 4.93°	0.30 m, 12.08°	0.21 m, 7.77°
MapNet+ [22]	0.10 m, 3.17°	0.20 m, 9.04°	0.13 m, 11.13°	0.18 m, 5.38°	0.19 m, 3.92°	0.20 m, 5.01°	0.30 m, 13.37°	0.19 m, 7.29°
DSAC++ [31]	0.02 m, 0.7°	0.03 m, 1.1°	0.12 m, 6.7°	0.03 m, 0.8°	0.05 m, 1.1°	0.05 m, 1.3°	0.29 m, 5.1°	0.084 m, 2.40°
LsG [95]	0.09 m, 3.28°	0.26 m, 10.92°	0.17 m, 12.70°	0.18 m, 5.45°	0.20 m, 3.69°	0.23 m, 4.92°	0.23 m, 11.3°	0.19 m, 7.47°
CAMNet [3]	0.04 m, 1.73°	0.03 m, 1.74°	0.05 m, 1.98°	0.04 m, 1.62°	0.04 m, 1.64°	0.04 m, 1.63°	0.04 m, 1.51°	0.040 m, 1.69°
CNN+GNN [56]	0.08 m, 2.82°	0.26 m, 8.94°	0.17 m, 11.41°	0.18 m, 5.08°	0.15 m, 2.77°	0.25 m, 4.48°	0.23 m, 8.78°	0.19 m, 6.33°
AtLoc [21]	0.10 m, 4.07°	0.25 m, 11.4°	0.16 m, 11.8°	0.17 m, 5.34°	0.21 m, 4.37°	0.23 m, 5.42°	0.26 m, 10.5°	0.20 m, 7.56°
KFNet [2]	0.019 m, 0.63°	0.023 m, 0.91°	0.018 m, 1.26°	0.026 m, 0.73°	0.039 m , 1.09°	0.039 m, 1.18°	0.037 m , 1.06°	0.029 m, 0.98°
DSM [26]	0.02 m, 0.71°	0.02 m, 0.85°	0.010 m, 0.85°	0.03 m, 0.84°	0.04 m, 1.16°	0.040 m, 1.17°	0.05 m, 1.33°	0.030 m, 0.987°
DSAC* [1]	0.019 m, 1.11°	0.019 m , 1.24°	0.011 m, 1.82°	0.026 m, 1.18°	0.042 m, 1.41°	0.03 m , 1.7°	0.041 m, 1.42°	0.027 m, 1.411°
TransPoseNet [89]	0.08 m, 5.68°	0.24 m, 10.6°	0.13 m, 12.7°	0.17 m, 6.34°	0.17 m, 5.6°	0.19 m, 6.75°	0.30 m, 7.02°	0.18 m, 7.81°
DFNet [96]	0.05 m, 1.88°	0.17 m, 6.45°	0.06 m, 3.63°	0.08 m, 2.48°	0.10 m, 2.78°	0.22 m, 5.45°	0.16 m, 3.29°	0.05 m, 1.64°
NeRF-IBVS [97]	0.03 m, 1.18°	0.03 m, 1.17°	0.02 m, 1.12°	0.04 m, 1.39°	0.08 m, 2.36°	0.07 m, 2.21°	0.08 m, 2.07°	0.05 m, 1.64°

Table 2. Cont.

Scene Scene Scale	Chess $3 \times 2 \text{ m}^2$	Fire $2.5 \times 1 \text{ m}^2$	Heads $2 \times 0.5 \text{ m}^2$	Office $2.5 \times 2 \text{ m}^2$	Pumpkin $2.5 \times 2 \text{ m}^2$	RedKitchen $4 \times 3 \text{ m}^2$	Stairs $2.5 \times 2 \text{ m}^2$	Avg.
NeuMap [57]	0.02 m, 0.81°	0.03 m, 1.11°	0.02 m, 1.17°	0.03 m, 0.98°	0.04 m, 1.11°	0.04 m, 1.33°	0.04 m, 1.12°	0.031 m, 1.09°
UBGL (ours)	0.018 m, 0.63°	0.019 m, 0.85°	0.011 m, 0.70°	0.025 m, 0.72°	0.039 m, 0.97°	0.041 m, 1.27°	0.037 m, 1.07°	0.027 m, 0.89°

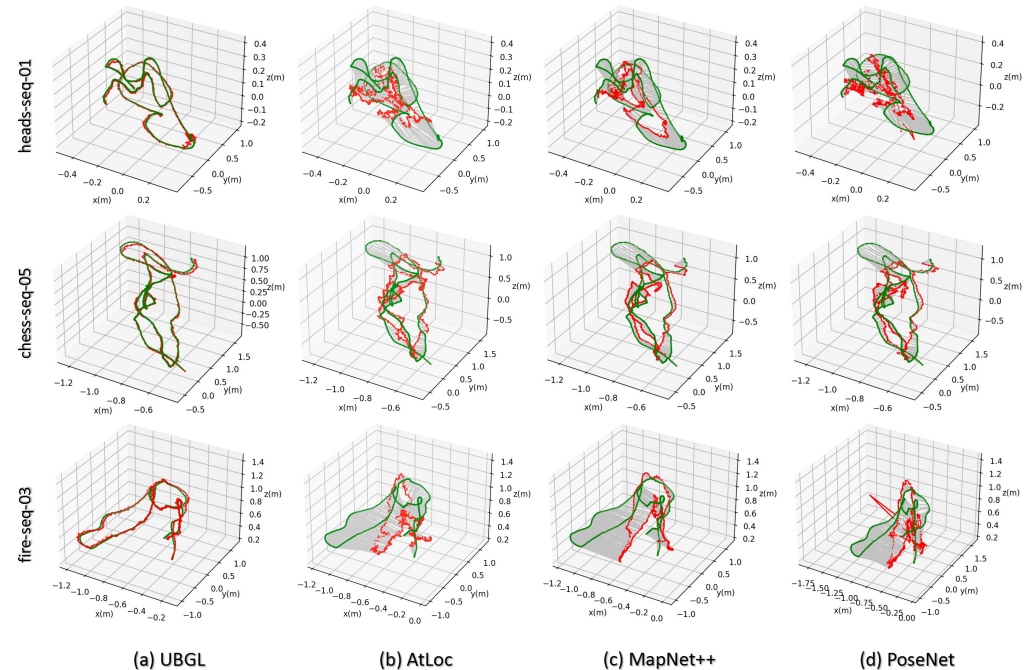


Figure 5. Visual comparison of camera trajectory estimation. We compared different models on the heads-seq-01, chess-seq-05, and fire-seq-03 sequences from the 7Scenes dataset [40]. The ground-truth trajectory is shown in green, while the predicted trajectory is shown in red.

4.2.2. Outdoor Localization Results on the Cambridge Benchmark Dataset [19]

Table 3 presents the comparison on the Cambridge Landmarks benchmark [19]. To avoid ambiguity caused by different averaging protocols in prior publications, we report both Avg-4 and Avg-5. Avg-4 was computed over KingsCollege, ShopFacade, StMarysChurch, and OldHospital, while Avg-5 additionally included GreatCourt when available.

Table 3. Evaluation on the Cambridge Landmarks outdoor localization benchmark [19]. The results of baseline methods are collected from their original publications. Avg-4 is computed over KingsCollege, ShopFacade, StMarysChurch, and OldHospital. Avg-5 additionally includes GreatCourt when the corresponding result is available. The leading entries under each averaging protocol are shown in bold. Because the compared methods may differ in input modalities, supervision, reference images, temporal information, or use of 3D information, the results should be interpreted as representative benchmark comparisons.

Scene Scene Scale	KingsCollege $5.6 \times 10^3 \text{ m}^2$	ShopFacade $8.8 \times 10^3 \text{ m}^2$	StMarysChurch $4.8 \times 10^3 \text{ m}^2$	OldHospital $2.0 \times 10^3 \text{ m}^2$	GreatCourt $8.0 \times 10^3 \text{ m}^2$	Avg-4 -	Avg-5 -
Active Search [91]	0.42 m, 0.6°	0.12 m, 0.4°	0.19 m, 0.54°	0.44 m, 1.0°	1.2 m, 0.6°	0.293 m, 0.635°	0.474 m, 0.628°
PoseNet17 [20]	0.99 m, 1.06°	1.05 m, 3.97°	1.49 m, 3.43°	2.17 m, 2.94°	7.00 m, 3.65°	1.425 m, 2.850°	2.540 m, 3.010°
DSAC [4]	0.3 m, 0.5°	0.09 m, 0.4°	0.55 m, 1.6°	0.33 m, 0.6°	2.8 m, 1.5°	0.318 m, 0.775°	0.814 m, 0.920°
InLoc [90]	0.46 m, 0.82°	0.11 m, 0.5°	0.18 m, 0.63°	0.48 m, 1.0°	1.2 m, 0.62°	0.308 m, 0.738°	0.486 m, 0.714°
MapNet [22]	1.07 m, 1.89°	1.49 m, 4.22°	2.00 m, 4.53°	1.94 m, 3.91°	7.85 m, 3.76°	1.625 m, 3.638°	2.870 m, 3.662°
DSAC++ [31]	0.23 m, 0.4°	0.09 m, 0.4°	0.29 m, 0.7°	0.24 m, 0.5°	0.66 m, 0.4°	0.213 m, 0.500°	0.302 m, 0.480°
SANet [25]	0.32 m, 0.42°	0.1 m, 0.47°	0.16 m, 0.57°	0.32 m, 0.53°	3.28 m, 1.95°	0.225 m, 0.498°	0.836 m, 0.788°
CNN+GNN [56]	0.59 m, 0.65°	0.50 m, 2.87°	1.90 m, 3.29°	1.88 m, 2.78°	6.67 m, 2.79°	1.218 m, 2.398°	2.308 m, 2.476°
KFNet [2]	0.16 m, 0.29°	0.05 m, 0.34°	0.12 m, 0.36°	0.18 m, 0.29°	0.43 m, 0.2°	0.128 m, 0.320°	0.188 m, 0.296°
DSM [26]	0.19 m, 0.36°	0.07 m, 0.38°	0.12 m, 0.35°	0.24 m, 0.39°	0.44 m, 0.23°	0.155 m, 0.370°	0.212 m, 0.342°

Table 3. Cont.

Scene Scene Scale	KingsCollege $5.6 \times 10^3 \text{ m}^2$	ShopFacade $8.8 \times 10^3 \text{ m}^2$	StMarysChurch $4.8 \times 10^3 \text{ m}^2$	OldHospital $2.0 \times 10^3 \text{ m}^2$	GreatCourt $8.0 \times 10^3 \text{ m}^2$	Avg-4 -	Avg-5 -
DSAC* [1]	0.179 m, 0.310°	0.052 m, 0.25°	0.15 m, 0.5°	0.212 m, 0.38°	0.335 m, 0.21°	0.148 m, 0.360°	0.186 m, 0.330°
TransPoseNet [89]	0.60 m, 2.43°	0.55 m, 3.49°	1.09 m, 4.94°	1.45 m, 3.08°	-, -	0.923 m, 3.485°	-, -
DFNet [96]	0.43 m, 0.87°	0.16 m, 0.59°	0.50 m, 1.49°	0.46 m, 0.87°	-, -	0.388 m, 0.955°	-, -
NeuMap [57]	0.14 m, 0.19°	0.06 m, 0.25°	0.17 m, 0.53°	0.19 m, 0.36°	0.06 m, 0.10°	0.140 m, 0.333°	0.124 m, 0.286°
UBGL (ours)	0.156 m, 0.256°	0.045 m, 0.247°	0.117 m, 0.35°	0.22 m, 0.40°	0.33 m, 0.18°	0.135 m, 0.313°	0.174 m, 0.287°

Under the Avg-4 protocol, UBGL obtains 0.135 m and 0.313°, achieving the best average rotation error and competitive average translation error among the compared methods. Compared with NeuMap [57], UBGL achieves slightly lower Avg-4 translation and rotation errors. Under the Avg-5 protocol, UBGL obtains 0.174 m and 0.287°, while NeuMap obtains 0.124 m and 0.286°. This shows that UBGL does not uniformly outperform NeuMap when GreatCourt is included, especially in translation accuracy. Therefore, we interpret the Cambridge results as demonstrating competitive outdoor localization performance rather than consistent dominance over existing methods. These results suggest that the proposed graph-based feature reasoning is beneficial for outdoor localization, but the improvement is scene-dependent. UBGL performs well on scenes such as ShopFacade and StMarysChurch, while NeuMap remains particularly strong on GreatCourt.

4.3. Ablation Study

4.3.1. Effectiveness of BGC and UGC

To evaluate the effectiveness of the proposed BGC, we conducted experiments by incrementally adding components to the baseline model, i.e., ResNet18-FCN. As shown in Table 4, the variant with MGP but without BGC can be regarded as the no cross-graph communication baseline. Adding BGC improves localization accuracy on all selected scenes, indicating that cross-scale graph communication contributes to the performance gain. Further adding UGC provides additional improvements or maintains the best results, showing the complementary role of uncertainty-aware relation modeling.

Table 4. Component analysis of the proposed method on the 7Scenes indoor benchmark [40] and the Cambridge Landmarks outdoor benchmark [19]. The reported values are obtained under different model variants, and the best-performing results are marked in bold. The arrows ↓ indicate that lower values are better.

Candidate				Complexity		7Scenes					Cambridge	
ResNet-18	MGP	BGC	UGC	GFLOPs ↓	Param. ↓	Chess	Fire	Heads	Office	KingsCollege	ShopFacade	StMarysChurch
✓				35.53 G	6.88 M	0.13 m, 4.16°	0.23 m, 8.12°	0.13 m, 7.57°	0.048 m, 1.6°	0.25 m, 0.43°	0.13 m, 0.49°	0.83 m, 2.27°
✓	✓			39.31 G	7.67 M	0.025 m, 0.80°	0.024 m, 1.0°	0.014 m, 0.90°	0.034 m, 1.0°	0.19 m, 0.27°	0.066 m, 0.26°	0.16 m, 0.41°
✓	✓	✓		39.36 G	10.83 M	0.019 m, 0.64°	0.020 m, 0.89°	0.011 m, 0.70°	0.028 m, 0.75°	0.158 m, 0.257°	0.050 m, 0.251°	0.118 m, 0.35°
✓	✓	✓	✓	39.37 G	13.06 M	0.018 m, 0.63°	0.019 m, 0.85°	0.011 m, 0.70°	0.025 m, 0.72°	0.156 m, 0.256°	0.045 m, 0.247°	0.117 m, 0.35°

Moreover, Table 5 further isolates the effect of the cross-graph interaction design. The baseline with ResNet-18 + MGP does not perform cross-graph communication, while the “Naïve” interaction variant uses simple cross-scale aggregation. Compared with these two settings, the proposed CAT-based interaction achieves better localization accuracy, demonstrating that explicit cross-graph attention is more effective than no interaction or simple aggregation.

4.3.2. Effectiveness of UBGL

We conducted additional comparisons to evaluate the effectiveness of the proposed uncertainty-aware bidirectional graph learning (UBGL) approach. As shown in Table 6, compared with the widely used non-local operation (denoted as Baseline + NL), the explicit

modeling strategy (denoted as Baseline + MGP) yields more reliable results on the selected scenes. These findings suggest that capturing high-order relationships among semantic graph nodes is useful for improving localization robustness. Such modeling helps alleviate the ambiguity in challenging camera localization tasks, although the gains remain scene-dependent.

Table 5. Effect of inter-graph interaction on localization performance. We evaluated different inter-graph interaction settings on the 7Scenes [40] and Cambridge Landmarks [19] datasets.

Scene		Baseline (ResNet-18 + MGP)	Baseline + Naïve Interaction	Baseline + CAT (Ours)
7Scenes	Chess	0.025 m, 0.80°	0.023 m, 0.74°	0.019 m, 0.64°
	Fire	0.024 m, 1.0°	0.021 m, 0.93°	0.020 m, 0.89°
	Heads	0.014 m, 0.9°	0.012 m, 0.81°	0.011 m, 0.70°
	Office	0.034 m, 1.0°	0.031 m, 0.83°	0.028 m, 0.72°
Cambridge	KingsCollege	0.19 m, 0.27°	0.179 m, 0.263°	0.158 m, 0.257°
	ShopFacade	0.066 m, 0.26°	0.058 m, 0.255°	0.050 m, 0.251°
	StMarysChurch	0.16 m, 0.41°	0.135 m, 0.384°	0.118 m, 0.35°

Table 6. Comparison of feature enhancement strategies. We evaluated different feature enhancement modules on the 7Scenes [40] and Cambridge Landmarks [19] benchmarks.

Scene		Baseline (ResNet-18 + FCN)	Baseline + NL	Baseline + MGP (Ours)
7Scenes	Chess	0.13 m, 4.16°	0.092 m, 1.62°	0.025 m, 0.8°
	Fire	0.23 m, 8.12°	0.163 m, 3.15°	0.024 m, 1.0°
	Heads	0.13 m, 7.57°	0.061 m, 2.23°	0.014 m, 0.9°
	Office	0.048 m, 1.6°	0.042 m, 1.40°	0.034 m, 1.0°
Cambridge	KingsCollege	0.25 m, 0.43°	0.23 m, 0.38°	0.19 m, 0.27°
	ShopFacade	0.13 m, 0.49°	0.11 m, 0.39°	0.066 m, 0.26°
	StMarysChurch	0.83 m, 2.27°	0.48 m, 1.63°	0.16 m, 0.41°

4.3.3. Key Parameter Analysis

We further investigated the sensitivity of several key hyperparameters in UBGL, including the number of graph nodes, the number of feature hierarchy levels, and the number of Monte Carlo samples in UGC. Table 7 analyzes different graph node settings at each scale level, which controls the capacity of graph representation. Table 8 studies the number of feature hierarchy levels L , which affects the depth of multi-scale graph reasoning. Table 9 evaluates the Monte Carlo sampling number T used in uncertainty-aware graph learning. In addition, Table 5 compares different cross-graph communication designs, including no interaction, naive interaction, and the proposed CAT-based interaction.

The results show that $K_3 = 32$, $K_2 = 16$, $K_1 = 8$, $L = 3$, and $T = 50$ provide stable performance in our experiments. Setting $L = 1$ leads to noticeable performance degradation, while increasing the number of hierarchies to $L = 4$ does not bring consistent improvement and may reduce accuracy in some cases. This suggests that excessive hierarchy depth can introduce overly fine-grained or redundant graph relations. Similarly, increasing the sampling number beyond $T = 50$ does not consistently improve performance, indicating that a moderate number of Monte Carlo samples is sufficient for stable uncertainty-aware graph learning.

4.3.4. Detailed Analysis of Computational Complexity

We analyzed the computational cost of the three main graph modules. For MGP, the dominant cost comes from assigning $S_l = H_l W_l$ spatial features to K_l graph nodes, resulting in approximately $\mathcal{O}(S_l K_l C)$ complexity at the l th feature level. For BGC, given a source graph $\mathcal{V}_s \in \mathbb{R}^{K_s \times C}$ and a target graph $\mathcal{V}_t \in \mathbb{R}^{K_t \times C}$, CAT computes the cross-graph attention matrix and transfers source information with complexity $\mathcal{O}(K_s K_t C)$. For UGC, the graph convolution cost is $\mathcal{O}(K_l^2 C)$, while Monte Carlo sampling with T samples introduces an additional sampling overhead proportional to $\mathcal{O}(TK_l^2)$. Since $K_l \in \{8, 16, 32\}$ is

small in our implementation, the graph reasoning and uncertainty sampling overheads are moderate compared with the CNN backbone.

Table 7. Effect of graph node number on localization performance. We compared different node settings in the graph representation, and the best-performing results are highlighted in bold.

Setting	7Scenes				Cambridge		
	Chess	Fire	Heads	Office	KingsCollege	ShopFacade	StMarysChurch
K3 = 16, K2 = 16, K1 = 8	0.0184 m, 0.64°	0.0202 m, 0.89°	0.0131 m, 0.76°	0.0254 m, 0.75°	0.158 m, 0.257°	0.049 m, 0.251°	0.118 m, 0.35°
K3 = 32, K2 = 16, K1 = 8	0.0183 m, 0.63°	0.0192 m, 0.85°	0.0109 m, 0.70°	0.0246 m, 0.72°	0.156 m, 0.256°	0.045 m, 0.247°	0.117 m, 0.35°
K3 = 64, K2 = 16, K1 = 8	0.0185 m, 0.63°	0.0188 m, 0.86°	0.0108 m, 0.71°	0.0255 m, 0.73°	0.158 m, 0.259°	0.045 m, 0.247°	0.117 m, 0.36°
K3 = 32, K2 = 8, K1 = 8	0.0185 m, 0.67°	0.0195 m, 0.87°	0.0107 m, 0.72°	0.0252 m, 0.75°	0.158 m, 0.260°	0.046 m, 0.258°	0.120 m, 0.35°
K3 = 32, K2 = 16, K1 = 8	0.0183 m, 0.63°	0.0192 m, 0.85°	0.0109 m, 0.70°	0.0246 m, 0.72°	0.156 m, 0.256°	0.045 m, 0.247°	0.117 m, 0.35°
K3 = 32, K2 = 32, K1 = 8	0.0182 m, 0.64°	0.0192 m, 0.88°	0.0117 m, 0.70°	0.0251 m, 0.72°	0.156 m, 0.257°	0.045 m, 0.247°	0.119 m, 0.36°
K3 = 32, K2 = 16, K1 = 4	0.0185 m, 0.64°	0.0195 m, 0.89°	0.0110 m, 0.71°	0.0267 m, 0.75°	0.157 m, 0.280°	0.050 m, 0.251°	0.118 m, 0.35°
K3 = 32, K2 = 16, K1 = 8	0.0183 m, 0.63°	0.0192 m, 0.85°	0.0109 m, 0.70°	0.0246 m, 0.72°	0.156 m, 0.256°	0.045 m, 0.247°	0.117 m, 0.35°
K3 = 32, K2 = 16, K1 = 16	0.0185 m, 0.62°	0.0192 m, 0.87°	0.0108 m, 0.71°	0.0251 m, 0.72°	0.156 m, 0.256°	0.046 m, 0.240°	0.118 m, 0.36°

Table 8. Parameter analysis on the 7Scenes [40] and Cambridge Landmarks [19] datasets. Here, L denotes the number of feature hierarchy levels used in the model.

Scene		$L = 1$	$L = 3$	$L = 4$
7Scenes	Chess	0.020 m, 0.68°	0.0183 m, 0.63°	0.019 m, 0.63°
	Fire	0.0279 m, 1.16°	0.0192 m, 0.85°	0.0192 m, 0.87°
	Heads	0.0144 m, 0.98°	0.0109 m, 0.70°	0.0110 m, 0.70°
	Office	0.0373 m, 1.04°	0.0246 m, 0.72°	0.0254 m, 0.72°
Cambridge	KingsCollege	0.179 m, 0.3°	0.156 m, 0.256°	0.161 m, 0.244°
	ShopFacade	0.065 m, 0.28°	0.045 m, 0.247°	0.049 m, 0.263°
	StMarysChurch	0.162 m, 0.52°	0.117 m, 0.35°	0.117 m, 0.36°

Table 9. Sensitivity study of the sample number. We evaluated different values of T on the 7Scenes [40] and Cambridge Landmarks [19] benchmarks, where T represents the number of Monte Carlo samples used in UGC.

Scene		$T = 10$	$T = 50$	$T = 100$
7Scenes	Chess	0.0185 m, 0.67°	0.0183 m, 0.63°	0.0185 m, 0.63°
	Fire	0.0196 m, 0.89°	0.0192 m, 0.85°	0.0193 m, 0.85°
	Heads	0.0110 m, 0.71°	0.0109 m, 0.70°	0.0108 m, 0.70°
	Office	0.0252 m, 0.73°	0.0246 m, 0.72°	0.0246 m, 0.73°
Cambridge	KingsCollege	0.158 m, 0.259°	0.156 m, 0.256°	0.156 m, 0.257°
	ShopFacade	0.049 m, 0.249°	0.045 m, 0.247°	0.046 m, 0.247°
	StMarysChurch	0.119 m, 0.36°	0.117 m, 0.35°	0.117 m, 0.35°

To provide a comprehensive analysis of efficiency, we report GFLOPs, parameter size, and accuracy for comparison. As shown in Table 4, compared with the backbone, our MGP module achieves notable improvements in localization accuracy while introducing only a modest increase in GFLOPs and parameters. This module effectively captures multi-scale semantic information, which is essential for the subsequent uncertainty-aware bidirectional graph reasoning with BGC and UGC. The fully equipped UBGL model requires 39.37 GFLOPs and 13.06 M parameters, maintaining competitive performance with moderate computational overhead.

During inference, the input image is passed through the backbone, graph modules, scene-coordinate head, and differentiable pose solver once. The additional graph cost is limited because graph reasoning operates on compact graph nodes rather than dense image pixels. Higher-resolution inputs mainly increase the cost of the CNN backbone and the graph projection/reprojection stages, while the graph reasoning cost remains mainly controlled by the node numbers. Since wall-clock inference time can vary with

GPU type and solver implementation, we report GFLOPs and parameter size as hardware-independent efficiency indicators.

4.3.5. Scalability

UBGL performs graph reasoning on compact semantic graph nodes rather than directly on all image pixels. For each feature level, the graph node number is small, i.e., $K_l \in \{8, 16, 32\}$, so the cost of graph communication is mainly determined by the graph node numbers instead of the original image resolution. Higher-resolution inputs mainly increase the cost of the CNN backbone and the graph projection/reprojection stages, while the graph reasoning cost remains limited. For larger and more complex scenes, UBGL can be applied under the same scene-specific training setting as existing scene-coordinate regression methods. However, memory usage and training cost may increase with scene diversity, image resolution, and the number of training images.

4.3.6. Meaning of Uncertainty Estimation

Uncertainty estimation plays an important role in camera localization because graph relations constructed from visual features can be unreliable in textureless regions, repetitive structures, occlusions, and motion-blurred areas. In our framework, UGC is not designed to simply emphasize uncertain intra-graph affinities. Instead, it models intra-graph relations probabilistically and reduces the dependence on a single deterministic affinity matrix. This allows the graph reasoning process to preserve stable relations while being less sensitive to noisy feature interactions. The sensitivity study in Table 9 shows that using $T = 50$ Monte Carlo samples provides stable localization performance in our setting.

To further validate the contribution of UGC, we summarize the corresponding component-analysis results in Table 10. Compared with the variant using MGP and BGC but without UGC, adding UGC improves or maintains localization accuracy across the selected 7Scenes and Cambridge Landmarks scenes. This supports that uncertainty-aware graph convolution contributes to the final performance gain. However, the current validation mainly demonstrates the empirical contribution of UGC through ablation and sensitivity analysis. It does not provide a full uncertainty-calibration evaluation or direct visualization of suppressed graph relations. Therefore, we interpret UGC as a probabilistic relation-modeling mechanism that improves robustness in our experiments, rather than claiming that the learned uncertainty is fully calibrated. More detailed uncertainty calibration and relation-level visualization will be investigated in future work.

Table 10. Contribution of uncertainty-aware graph convolution. The results are directly summarized from the component analysis in Table 4. Compared with the variant without UGC, adding UGC improves or maintains localization accuracy across the selected scenes.

Variant	7Scenes				Cambridge Landmarks		
	Chess	Fire	Heads	Office	KingsCollege	ShopFacade	StMarysChurch
ResNet-18 + MGP + BGC, w/o UGC	0.019 m, 0.64°	0.020 m, 0.89°	0.011 m, 0.70°	0.028 m, 0.75°	0.158 m, 0.257°	0.050 m, 0.251°	0.118 m, 0.35°
ResNet-18 + MGP + BGC + UGC	0.018 m, 0.63°	0.019 m, 0.85°	0.011 m, 0.70°	0.025 m, 0.72°	0.156 m, 0.256°	0.045 m, 0.247°	0.117 m, 0.35°

4.3.7. Error Characterization and Failure Cases

Although median translation and rotation errors are the standard metrics for camera localization benchmarks, they do not fully describe the distribution of localization failures. Since the current benchmark tables mainly follow the commonly used median-error protocol, we further provide a qualitative error-characterization discussion. Larger localization errors are typically associated with severe motion blur, weak or textureless regions, repetitive structures, partial occlusions, and large viewpoint changes. These factors can make local visual correspondences ambiguous and lead to unstable scene-coordinate predictions.

For indoor scenes such as 7Scenes, failure cases are often related to motion blur, reflective surfaces, and repeated structures in small rooms. For outdoor scenes such as Cambridge Landmarks, large viewpoint changes, illumination variations, and insufficient coverage of certain scene regions can increase localization errors. The uncertainty-aware graph module helps reduce the influence of unreliable feature interactions, but it cannot completely eliminate failures caused by severe visual ambiguity or limited scene coverage. The trajectory visualization in Figure 5 also indicates that most predictions follow the ground-truth trajectory, while larger deviations tend to appear in visually ambiguous frames.

5. Conclusions

5.1. Main Findings

In this paper, we proposed an uncertainty-aware bidirectional graph learning framework, termed UBGL, for end-to-end camera localization. Specifically, a multi-scale graph projection (MGP) module was introduced to effectively capture rich semantic representations, followed by uncertainty-aware bidirectional graph communication (BGC and UGC) to model contextual relationships and mitigate ambiguity. Our method provides a new perspective on graph-based learning by incorporating Bayesian principles into graph reasoning. Extensive experiments demonstrated that UBGL achieved competitive performance across indoor and outdoor benchmarks, with strong rotation accuracy on 7Scenes and comparable localization accuracy on Cambridge Landmarks, while maintaining a favorable efficiency–accuracy trade-off.

5.2. Limitations

Although UBGL achieves competitive performance on standard indoor and outdoor localization benchmarks, it may still face challenges in highly dynamic environments and very large outdoor scenes. Moving objects, long-term appearance changes, illumination variations, seasonal changes, and insufficient scene coverage can affect graph relation estimation and scene-coordinate prediction. The uncertainty-aware graph module can reduce the influence of unstable feature interactions to some extent, but it does not fully solve the problem of dynamic scene changes or large-scale scene generalization. In addition, this work evaluates UBGL on 7Scenes and Cambridge Landmarks, which are widely used in scene-coordinate-based camera localization. Evaluation on larger-scale benchmarks such as Aachen Day-Night or RobotCar would further demonstrate robustness and generalization under more challenging long-term outdoor conditions. We leave this extension as an important direction for future work.

Author Contributions: Conceptualization, H.C., X.L. and Q.Z.; methodology, H.C. and Y.C.; software, H.C.; validation, H.C. and Z.L.; formal analysis, H.C.; investigation, H.C.; resources, X.L. and Q.Z.; data curation, H.C. and Y.C.; writing—original draft preparation, H.C.; writing—review and editing, Y.C., Z.L., F.Y., X.L. and Q.Z.; visualization, H.C.; supervision, Z.L., X.L. and Q.Z.; project administration, Z.L.; funding acquisition, X.L. and Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the General Program of the Natural Science Foundation of Sichuan Province, grant number 2025ZNSFSC0162, and the Chongqing Social Science Planning Doctoral Project, grant number 2024BS043. The APC was funded by these grants.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to the use of publicly available benchmark datasets that do not involve human participants or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new datasets were created in this study. The 7Scenes and Cambridge Landmarks datasets used in this work are publicly available from their original sources. The experiments in this paper were conducted only on these two datasets.

Conflicts of Interest: Author Yan Cheng was employed by the company PipeChina Northwest Branch and author Fan Yang and Xin Li were employed by the company AIQ. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Brachmann, E.; Rother, C. Visual camera re-localization from rgb and rgb-d images using dsac. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 5847–5865.
2. Zhou, L.; Luo, Z.; Shen, T.; Zhang, J.; Zhen, M.; Yao, Y.; Fang, T.; Quan, L. Kfnet: Learning temporal camera relocalization using kalman filtering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4919–4928.
3. Ding, M.; Wang, Z.; Sun, J.; Shi, J.; Luo, P. CamNet: Coarse-to-fine retrieval for camera re-localization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 2871–2880.
4. Brachmann, E.; Krull, A.; Nowozin, S.; Shotton, J.; Michel, F.; Gumhold, S.; Rother, C. Dsac-differentiable ransac for camera localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6684–6692.
5. Clark, R.; Wang, S.; Markham, A.; Trigoni, N.; Wen, H. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6856–6864.
6. Duh, P.J.; Sung, Y.C.; Chiang, L.Y.F.; Chang, Y.J.; Chen, K.W. V-eye: A vision-based navigation system for the visually impaired. *IEEE Trans. Multimed.* **2020**, *23*, 1567–1580.
7. Song, Y.; Chen, X.; Wang, X.; Zhang, Y.; Li, J. 6-DOF image localization from massive geo-tagged reference images. *IEEE Trans. Multimed.* **2016**, *18*, 1542–1554. [[CrossRef](#)]
8. Yang, B.; Ran, W.; Wang, L.; Lu, H.; Chen, Y.P.P. Multi-Classes and Motion Properties for Concurrent Visual SLAM in Dynamic Environments. *IEEE Trans. Multimed.* **2021**, *24*, 3947–3960. [[CrossRef](#)]
9. Li, Y.; Fu, C.; Huang, Z.; Zhang, Y.; Pan, J. Intermittent contextual learning for keyfilter-aware uav object tracking using deep convolutional feature. *IEEE Trans. Multimed.* **2020**, *23*, 810–822. [[CrossRef](#)]
10. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [[CrossRef](#)]
11. Salarian, M.; Iliev, N.; Cetin, A.E.; Ansari, R. Improved image-based localization using sfm and modified coordinate system transfer. *IEEE Trans. Multimed.* **2018**, *20*, 3298–3310. [[CrossRef](#)]
12. Yang, X.; Yuan, Z.; Zhu, D.; Chi, C.; Li, K.; Liao, C. Robust and Efficient RGB-D SLAM in Dynamic Environments. *IEEE Trans. Multimed.* **2020**, *23*, 4208–4219. [[CrossRef](#)]
13. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.; Tardós, J.D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimodal SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [[CrossRef](#)]
14. Luo, H.; Gao, Y.; Wu, Y.; Liao, C.; Yang, X.; Cheng, K.T. Real-time dense monocular SLAM with online adapted depth prediction network. *IEEE Trans. Multimed.* **2018**, *21*, 470–483. [[CrossRef](#)]
15. Piao, J.C.; Kim, S.D. Real-time visual-inertial SLAM based on adaptive keyframe selection for mobile AR applications. *IEEE Trans. Multimed.* **2019**, *21*, 2827–2836. [[CrossRef](#)]
16. Yang, N.; Stumberg, L.v.; Wang, R.; Cremers, D. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1281–1292.
17. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 225–234.
18. Xiao, Y.; Yuan, Q.; Jiang, K.; He, J.; Lin, C.W.; Zhang, L. TTST: A top-k token selective transformer for remote sensing image super-resolution. *IEEE Trans. Image Process.* **2024**, *33*, 738–752. [[CrossRef](#)] [[PubMed](#)]
19. Kendall, A.; Grimes, M.; Cipolla, R. PoseNet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2938–2946.
20. Kendall, A.; Cipolla, R. Geometric loss functions for camera pose regression with deep learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5974–5983.

21. Wang, B.; Chen, C.; Lu, C.X.; Zhao, P.; Trigoni, N.; Markham, A. Atloc: Attention guided camera localization. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 10393–10401.
22. Brahmabhatt, S.; Gu, J.; Kim, K.; Hays, J.; Kautz, J. Geometry-aware learning of maps for camera localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2616–2625.
23. Huang, Z.; Xu, Y.; Shi, J.; Zhou, X.; Bao, H.; Zhang, G. Prior guided dropout for robust visual localization in dynamic environments. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2791–2800.
24. Sattler, T.; Zhou, Q.; Pollefeys, M.; Leal-Taixe, L. Understanding the limitations of cnn-based absolute camera pose regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3302–3312.
25. Yang, L.; Bai, Z.; Tang, C.; Li, H.; Furukawa, Y.; Tan, P. SANet: Scene agnostic network for camera localization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 42–51.
26. Tang, S.; Tang, C.; Huang, R.; Zhu, S.; Tan, P. Learning Camera Localization via Dense Scene Matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1831–1841.
27. Wang, S.; Laskar, Z.; Melekhov, I.; Li, X.; Zhao, Y.; Tolias, G.; Kannala, J. Hscnet++: Hierarchical scene coordinate classification and regression for visual localization with transformer. *Int. J. Comput. Vis.* **2024**, *132*, 2530–2550. [[CrossRef](#)]
28. Revaud, J.; Cabon, Y.; Brégier, R.; Lee, J.; Weinzaepfel, P. Sacreg: Scene-agnostic coordinate regression for visual localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 688–698.
29. Jiang, X.; Wang, F.; Galliani, S.; Vogel, C.; Pollefeys, M. R-score: Revisiting scene coordinate regression for robust large-scale visual localization. In Proceedings of the Computer Vision and Pattern Recognition Conference, Nashville, TN, USA, 15–18 October 2025; pp. 11536–11546.
30. Mao, X.; Li, H.; Wu, Y.; Li, J.; Bai, X.; Zheng, J. CoLoR: The Devil is in Scene Coordinate Regression for Large-Scale Visual Localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Denver, CO, USA, 3–7 June 2026; pp. 12206–12216.
31. Brachmann, E.; Rother, C. Learning less is more-6d camera localization via 3d surface regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4654–4662.
32. Li, X.; Yang, F.; Cheng, H.; Liu, W.; Shen, D. Contour knowledge transfer for salient object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 355–370.
33. Li, X.; Yang, F.; Cheng, H.; Chen, J.; Guo, Y.; Chen, L. Multi-scale cascade network for salient object detection. In Proceedings of the 25th ACM International Conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 439–447.
34. Yang, F.; Li, X.; Cheng, H.; Li, J.; Chen, L. Object-aware dense semantic correspondence. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–27 July 2017; pp. 2777–2785.
35. Yang, F.; Li, X.; Cheng, H.; Guo, Y.; Chen, L.; Li, J. Multi-scale bidirectional fcn for object skeleton extraction. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
36. Yang, F.; Li, X.; Shen, J. MSB-FCN: Multi-scale bidirectional fcn for object skeleton extraction. *IEEE Trans. Image Process* **2020**, *30*, 2301–2312. [[CrossRef](#)]
37. Kendall, A.; Badrinarayanan, V.; Cipolla, R. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv* **2015**, arXiv:1511.02680.
38. Kendall, A.; Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv* **2017**, arXiv:1703.04977.
39. Wang, H.; Yeung, D.Y. A survey on Bayesian deep learning. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–37. [[CrossRef](#)]
40. Shotton, J.; Glocker, B.; Zach, C.; Izadi, S.; Criminisi, A.; Fitzgibbon, A. Scene coordinate regression forests for camera relocation in RGB-D images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2930–2937.
41. Ma, J.; Jiang, X.; Fan, A.; Jiang, J.; Yan, J. Image matching from handcrafted to deep features: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 23–79.
42. Sualeh, M.; Kim, G.W. Simultaneous localization and mapping in the epoch of semantics: A survey. *Int. J. Control. Autom. Syst.* **2019**, *17*, 729–742.
43. Barnett, J.; Gizinski, N.; Mondragon-Parra, E.; Siegel, J.E.; Morris, D.; Gates, T.; Kassens-Noor, E.; Savolainen, P. Automated vehicles sharing the road: Surveying detection and localization of pedalcyclists. *IEEE Trans. Intell. Veh.* **2020**, *6*, 649–664. [[CrossRef](#)]
44. Sato, R. A survey on the expressive power of graph neural networks. *arXiv* **2020**, arXiv:2003.04078.
45. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [[CrossRef](#)]

46. Balntas, V.; Li, S.; Prisacariu, V. Relocnet: Continuous metric learning relocalisation using neural nets. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 751–767.
47. Laskar, Z.; Melekhov, I.; Kalia, S.; Kannala, J. Camera relocalization by computing pairwise relative poses using convolutional neural network. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 929–938.
48. Saha, S.; Varma, G.; Jawahar, C.V. Improved Visual Relocalization by Discovering Anchor Points. In Proceedings of the Proceedings of British Machine Vision Conference, Newcastle, UK, 3–6 September 2018; p. 164.
49. Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J. NetVLAD: CNN architecture for weakly supervised place recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5297–5307.
50. Zhai, Q.; Huang, R.; Cheng, H.; Zhan, H.; Li, J.; Liu, Z. Learning Quintuplet Loss for Large-Scale Visual Geolocalization. *IEEE Multimed.* **2020**, *27*, 34–43. [[CrossRef](#)]
51. Torii, A.; Arandjelovic, R.; Sivic, J.; Okutomi, M.; Pajdla, T. 24/7 place recognition by view synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–15 June 2015; pp. 1808–1817.
52. Zhu, Y.; Wang, J.; Xie, L.; Zheng, L. Attention-based pyramid aggregation network for visual place recognition. In Proceedings of the 26th ACM International Conference on Multimedia, Seoul, Republic of Korea, 26 October 2018; pp. 99–107.
53. Kendall, A.; Cipolla, R. Modelling uncertainty in deep learning for camera relocalization. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 4762–4769.
54. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 4015–4026.
55. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. DINOv2: Learning robust visual features without supervision. *arXiv* **2023**, arXiv:2304.07193.
56. Xue, F.; Wu, X.; Cai, S.; Wang, J. Learning multi-view camera relocalization with graph neural networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11372–11381.
57. Tang, S.; Tang, S.; Tagliasacchi, A.; Tan, P.; Furukawa, Y. Neumap: Neural coordinate mapping by auto-transdecoder for camera localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2023; pp. 929–939.
58. Zhai, Q.; Li, X.; Yang, F.; Chen, C.; Cheng, H.; Fan, D.P. Mutual graph learning for camouflaged object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 12997–13007.
59. Yang, F.; Zhai, Q.; Li, X.; Huang, R.; Cheng, H.; Fan, D.P. Uncertainty-Guided Transformer Reasoning for Camouflaged Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021.
60. Luo, A.; Li, X.; Yang, F.; Jiao, Z.; Cheng, H.; Lyu, S. Cascade graph neural networks for rgb-d salient object detection. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 346–364.
61. Luo, A.; Yang, F.; Li, X.; Nie, D.; Jiao, Z.; Zhou, S.; Cheng, H. Hybrid graph neural networks for crowd counting. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11693–11700.
62. Wei, X.; Yu, R.; Sun, J. View-GCN: View-based graph convolutional network for 3D shape analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1850–1859.
63. Zhai, Q.; Yang, F.; Li, X.; Xie, G.S.; Cheng, H.; Liu, Z. Co-communication graph convolutional network for multi-view crowd counting. *IEEE Trans. Multimed.* **2022**, *25*, 5813–5825.
64. Zhang, Y.; Deng, W.; Wang, M.; Hu, J.; Li, X.; Zhao, D.; Wen, D. Global-local gcn: Large-scale label noise cleansing for face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7731–7740.
65. Mohamed, A.; Qian, K.; Elhoseiny, M.; Claudel, C. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 14424–14432.
66. Yang, H.; Zhen, X.; Chi, Y.; Zhang, L.; Hua, X.S. Cpr-gcn: Conditional partial-residual graph convolutional network in automated anatomical labeling of coronary arteries. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3803–3811.
67. Xie, G.S.; Liu, L.; Zhu, F.; Zhao, F.; Zhang, Z.; Yao, Y.; Qin, J.; Shao, L. Region graph embedding network for zero-shot learning. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 562–580.
68. Nie, W.; Ren, M.; Liu, A.; Mao, Z.; Nie, J. M-GCN: Multi-branch graph convolution network for 2D image-based on 3D model retrieval. *IEEE Trans. Multimed.* **2020**, *23*, 1962–1976.

69. Luo, A.; Yang, F.; Luo, K.; Li, X.; Fan, H.; Liu, S. Learning optical flow with adaptive graph reasoning. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 22 February–1 March 2022; Volume 36, pp. 1890–1898.
70. Wang, W.; Gao, J.; Yang, X.; Xu, C. Learning coarse-to-fine graph neural networks for video-text retrieval. *IEEE Trans. Multimed.* **2020**, *23*, 2386–2397.
71. Zhang, L.; Chen, L.; Zhou, C.; Yang, F.; Li, X. Exploring graph-structured semantics for cross-modal retrieval. In Proceedings of the 29th ACM International Conference on Multimedia, Chengdu, China, 20–24 October 2021; pp. 4277–4286.
72. Zhang, L.; Chen, L.; Zhou, C.; Li, X.; Yang, F.; Yi, Z. Weighted graph-structured semantics constraint network for cross-modal retrieval. *IEEE Trans. Multimed.* **2023**, *26*, 1551–1564.
73. Zheng, T.; Yao, H.; Zhao, S.; Jiang, K.; Xiao, Y. GraphMamba: Whole slide image classification meets graph-driven selective state space model. *Pattern Recognit.* **2025**, *167*, 111768. [[CrossRef](#)]
74. Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; Wierstra, D. Weight uncertainty in neural network. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1613–1622.
75. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1050–1059.
76. Ritter, H.; Botev, A.; Barber, D. A scalable laplace approximation for neural networks. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings, Vancouver, BC, Canada, 30 April–3 May 2018; Volume 6.
77. Maddox, W.J.; Izmailov, P.; Garipov, T.; Vetrov, D.P.; Wilson, A.G. A simple baseline for bayesian uncertainty in deep learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 13153–13164.
78. Li, X.; Fan, D.P.; Yang, F.; Luo, A.; Cheng, H.; Liu, Z. Probabilistic Model Distillation for Semantic Correspondence. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7505–7514.
79. Li, A.; Zhang, J.; Lv, Y.; Liu, B.; Zhang, T.; Dai, Y. Uncertainty-aware joint salient object and camouflaged object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10071–10081.
80. Yu, T.; Li, D.; Yang, Y.; Hospedales, T.M.; Xiang, T. Robust person re-identification by modelling feature uncertainty. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October 2019; pp. 552–561.
81. Huang, P.Y.; Hsu, W.T.; Chiu, C.Y.; Wu, T.F.; Sun, M. Efficient uncertainty estimation for semantic segmentation in videos. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 520–535.
82. Kwon, Y.; Won, J.H.; Kim, B.J.; Paik, M.C. Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation. *Comput. Stat. Data Anal.* **2020**, *142*, 106816. [[CrossRef](#)]
83. Zhang, H.; Xue, J.; Dana, K. Deep ten: Texture encoding network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 June 2017; pp. 708–717.
84. Li, Y.; Gupta, A. Beyond grids: Learning graph representations for visual recognition. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 9225–9235.
85. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
86. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
87. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
88. Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A.; et al. KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In Proceedings of the 24th annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, 16–19 October 2011; pp. 559–568.
89. Shavit, Y.; Ferens, R.; Keller, Y. Learning multi-scene absolute pose regression with transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 2733–2742.
90. Taira, H.; Okutomi, M.; Sattler, T.; Cimpoi, M.; Pollefeys, M.; Sivic, J.; Pajdla, T.; Torii, A. InLoc: Indoor visual localization with dense matching and view synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7199–7209.
91. Sattler, T.; Leibe, B.; Kobbelt, L. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1744–1756. [[PubMed](#)]
92. Walch, F.; Hazirbas, C.; Leal-Taixe, L.; Sattler, T.; Hilsenbeck, S.; Cremers, D. Image-based localization using lstms for structured feature correlation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 627–637.
93. Melekhov, I.; Ylioinas, J.; Kannala, J.; Rahtu, E. Image-based localization using hourglass networks. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 879–886.

94. Wu, J.; Ma, L.; Hu, X. Delving deeper into convolutional neural networks for camera relocalization. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5644–5651.
95. Xue, F.; Wang, X.; Yan, Z.; Wang, Q.; Wang, J.; Zha, H. Local supports global: Deep camera relocalization with sequence enhancement. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2841–2850.
96. Chen, S.; Li, X.; Wang, Z.; Prisacariu, V.A. Dfnet: Enhance absolute pose regression with direct feature matching. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–17.
97. Wang, Y.; Yan, Y.; Shi, D.; Zhu, W.; Xia, J.; Jeff, T.; Jin, S.; Gao, K.; Li, X.; Yang, X. NeRF-IBVS: Visual Servo Based on NeRF for Visual Localization and Navigation. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 8292–8304.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.