DGC-GNN: Leveraging Geometry and Color Cues for Visual Descriptor-Free 2D-3D Matching

Shuzhe Wang*   Juho Kannala*   Daniel Barath†

* Department of Computer Science, Aalto University
† Computer Vision and Geometry Group, ETH Zurich

shuzhe.wang@aalto.fi   juho.kannala@aalto.fi   dbarath@inf.ethz.ch

Abstract

Matching 2D keypoints in an image to a sparse 3D point cloud of the scene without requiring visual descriptors has garnered increased interest due to its low memory requirements, inherent privacy preservation, and reduced need for expensive 3D model maintenance compared to visual descriptor-based methods. However, existing algorithms often compromise on performance, resulting in a significant deterioration compared to their descriptor-based counterparts. In this paper, we introduce DGC-GNN, a novel algorithm that employs a global-to-local Graph Neural Network (GNN) that progressively exploits geometric and color cues to represent keypoints, thereby improving matching accuracy. Our procedure encodes both Euclidean and angular relations at a coarse level, forming the geometric embedding to guide the point matching. We evaluate DGC-GNN on both indoor and outdoor datasets, demonstrating that it not only doubles the accuracy of the state-of-the-art visual descriptor-free algorithm but also substantially narrows the performance gap between descriptor-based and descriptor-free methods.[1]

1. Introduction

Establishing 2D-3D matches plays a crucial role in various computer vision applications, including visual localization [19, 34, 35, 37, 39, 50], 3D reconstruction [7, 24, 40, 45], and Simultaneous Localization and Mapping (SLAM) [14, 28, 29]. Traditional methods for establishing point-to-point matches involve extracting keypoints and descriptors from a query image, then matching the 2D and 3D descriptors using exhaustive search. To circumvent the computationally expensive matching process, some approaches [19, 34] narrow the search space by employing image retrieval methods [1, 31] first to identify the most

---

[1]The code and trained models are available at: https://github.com/AaltoVision/DGC-GNN-release.
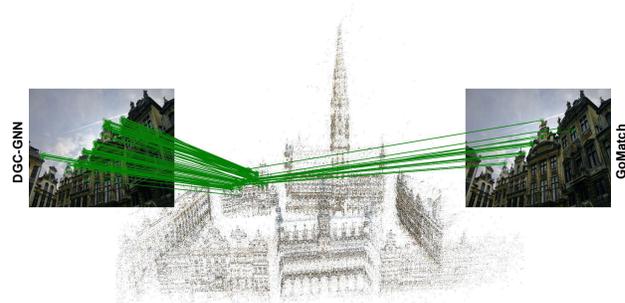

Figure 1. **2D-3D matching** (shown by green lines) with the proposed DGC-GNN and GoMatch [53]. In this example, DGC-GNN obtains 78 correct matches with 0.02 meters camera translation and 0.24° rotation errors, while GoMatch finds only 17 inliers with a pose error of 0.37 meters and 4.37°.

similar images in the database, and then perform descriptor-based image matching [10, 13, 26, 35, 42] between the query and retrieved images. The 2D-3D correspondences are subsequently established by connecting the 2D-2D image matches with the prebuilt 2D-3D correspondences in the database. Another approach [36] is to build 2D-to-3D matches by searching through all point descriptors with an efficient vocabulary-based method. Sattler *et al.* [37, 38] further explore the combination of both 2D-3D and 3D-2D search as an active correspondence search step for a faster and more efficient matching process.

While descriptor-based algorithms achieve state-of-the-art accuracy, they store and maintain high-dimensional visual descriptors for each point in potentially large 3D point clouds. The stored model often requires orders of magnitude more storage than the point cloud and images alone [53]. These methods are susceptible to privacy attacks [6, 11, 12, 32] and necessitate computationally expensive model maintenance and descriptor update procedures [53] when incorporating new descriptors or points into the model. Several approaches have been proposed to address these limitations. Yang *et al.* [51] employ learned point selection to sample a subset of

the point cloud for scene compression. Other methods [3, 22] directly learn a function that maps 2D pixels to 3D coordinates without explicitly storing the 3D scene. Additionally, [30] introduces an adversarial learning framework to develop content-concealing descriptors that prevent privacy leakage.

Recently, researchers [4, 25] have begun exploring deep learning techniques for cross-domain direct 2D-3D matching and pose estimation without visual descriptors, showcasing the potential of descriptor-free matching through differentiable geometric optimization. The recently proposed Go-Match [53] represents significant progress in descriptor-free 2D-3D keypoints matching, achieving reasonable matching performance on a variety of real-world datasets [21, 23, 43]. GoMatch first identifies keypoints in the query image, which, along with the 3D points from the model, are converted to bearing vectors in the camera coordinate system. The algorithm employs an attention mechanism [35, 49] to establish reliable 2D-3D correspondences effectively. While GoMatch attains reasonable accuracy, its performance still significantly lags behind its descriptor-based counterparts [34, 35, 38]. Additionally, it relies on geometric cues only from the points and their local neighbors, rendering it incapable of distinguishing geometrically similar structures.

These observations lead us to two critical questions: (1) Is geometry the only information we can utilize? (2) How can we leverage the geometric information derived from the points for matching? In practice, humans identify correspondences between objects by considering global structures and local geometric cues. For example, when matching an image to a point cloud as in Fig. 1, we first locate the building based on its unique structure and then identify the local structure of the roof for matching. Besides geometric cues, the visual context, such as the color information at each point, also provides constraints for 2D-3D matching. Importantly, this color information still preserves privacy, as the RGB data from sparse keypoints is insufficient to reconstruct the scene.

Building upon these observations and the groundwork set by GoMatch, we propose a novel graph-based pipeline, named **DGC-GNN**, which leverages geometric and color cues in a global-to-local manner for descriptor-free 2D-3D matching. DGC-GNN encodes position and RGB information for each point and extracts a global *distance-angular* embedding to guide local point matching. Taking inspiration from [42], we employ a cluster-based transformer to constrain information flow within local clusters. We observe, from real-world datasets, that DGC-GNN leads to substantial improvements in the number of correct matches and the accuracy of pose estimation. Notably, it *doubles the accuracy* of GoMatch, thereby reducing the gap between descriptor-based and descriptor-free methods. In summary, our paper makes the following contributions:

- We introduce a visual descriptor-free global-to-local GNN for direct 2D-3D keypoint matching. The network leverages multiple cues and incorporates a progressive clustering module to represent the keypoints. This pipeline enhances the accuracy of sparse 2D-3D matching while requiring low memory, being privacy-preserving, and low cost from 3D model maintenance.
- We demonstrate that color information for each point is crucial for 2D-3D matching. By incorporating RGB encoding into our network, we observe significant performance improvements.
- Extensive experiments on real-world datasets show that DGC-GNN outperforms previous methods by a large margin on both matching and visual localization tasks.

2. Visual Descriptor-Free 2D-3D Matching

2.1. Problem Formulation and Notation

Given keypoints $\mathbf{P} = \{\mathbf{p}_n \in \mathbb{R}^2 \mid n = 1, ..., N\}$ from query image I and database 3D point cloud $\mathbf{Q} = \{\mathbf{q}_m \in \mathbb{R}^3 \mid m = 1, ..., M\}$, where, optionally, each 3D point is associated with a visual descriptor $\mathbf{d} \in \mathbb{R}^D$. The task is to find a set $\mathcal{M}_{\mathbf{p},\mathbf{q}}$ of corresponding keypoints such that

$$\mathcal{M}_{\mathbf{p},\mathbf{q}} = \{(n, m) \mid ||\pi(\mathbf{q}_m, \mathbf{R}, \mathbf{t}, \mathbf{K}) - \mathbf{p}_n||_2 \leq \epsilon\}, \quad (1)$$

where $\pi(\cdot)$ is a mapping that projects a 3D point $\mathbf{q}_m$ from world coordinates to the image plane, represented by a camera rotation $\mathbf{R} \in \mathbb{R}^{3\times3}$, translation $\mathbf{t} \in \mathbb{R}^3$, and intrinsic parameter matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$. Parameter $\epsilon \in \mathbb{R}$ is the threshold specified in pixels. Additionally, we denote the color of point $\mathbf{p}_n$ as $\mathbf{c}_n = [r, g, b]^T \in [0, 1]^3$.

Bearing Vector. Similar to [53], we adopt bearing vectors as keypoint representation for both the 2D and 3D points to alleviate their cross-domain nature and represent them in the same space. The bearing vector is the direction from the camera center to a 3D point in the camera coordinate system. Given an image, a 2D pixel $\mathbf{p}_n$ is uplifted to bearing vector as $[\mathbf{b}_{\mathbf{p},n}, 1]^T = \mathbf{K}^{-1}[\mathbf{p}_n, 1]^T, \mathbf{b}_{\mathbf{p},n} \in \mathbb{R}^2$, where $\mathbf{K}$ is the intrinsic camera matrix. Given a 3D point $\mathbf{q}_m$, the corresponding bearing vector is

$$[\mathbf{b}_{\mathbf{q},m}, 1]^T = \frac{\mathbf{R}\mathbf{q}_m + \mathbf{t}}{[\mathbf{R}\mathbf{q}_m + \mathbf{t}]_z}, \quad (2)$$

where $\mathbf{R}$ is the camera rotation and $\mathbf{t}$ is its translation in the world coordinate system and subscript z denotes the third component of the 3D vector.

2.2. Network Architecture

The proposed DGC-GNN applies a hierarchical mechanism to leverage color and geometric cues in a global-to-local fashion. The overall pipeline is illustrated in Fig.2. We initially employ two local feature extractors to encode RGB and position information for each point simultaneously (Sec. 2.2.1). Additionally, we cluster the points based on their distances
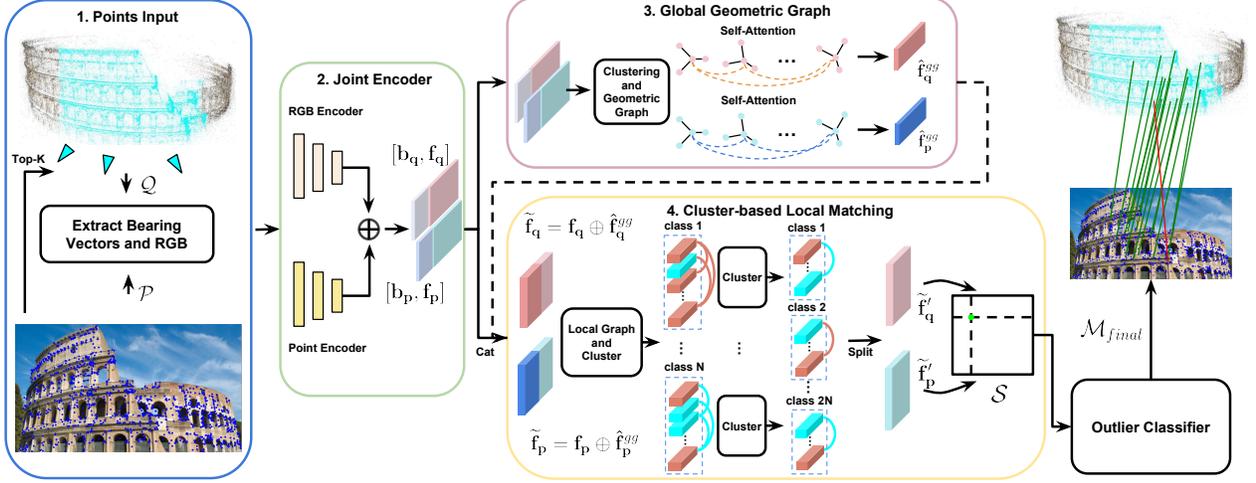
Figure 2. **Pipeline overview.** For keypoints from the 2D image and 3D points from the point cloud, the proposed DGC-GNN (1) considers the bearing vectors and the color at each bearing vector as input. (2) It extracts the point-wise position and color features with two separate encoders and mixes the features as $\mathbf{f_p}$ and $\mathbf{f_q}$. (3) The bearing vectors are clustered into K groups, and geometric graphs are built upon the clusters to extract the global-level geometric embeddings $\hat{\mathbf{f}}_{\mathbf{p}}^{gg}$ and $\hat{\mathbf{f}}_{\mathbf{q}}^{gg}$. (4) We then concatenate $\hat{\mathbf{f}}_{\mathbf{p}}^{gg}$ with $\mathbf{f_p}$ and $\hat{\mathbf{f}}_{\mathbf{q}}^{gg}$ with $\mathbf{f_p}$, and build a local graph at each point as self-attention. A cluster-based attention module is adopted to enhance the local features by forcing the message to pass only with the most related features. A differentiable layer matches and optimizes the improved features to obtain score matrix $\mathcal{S}$. Finally, an outlier rejection network is applied to prune the matches with low confidence, leading to the final 2D-3D correspondences $\mathcal{M}_{final}$.

and generate global graphs to obtain the global-level geometric embeddings (Sec. 2.2.2). Next, we concatenate the local point features with their corresponding global features and input them into the cluster-based local matching module to identify the initial matches (Sec. 2.2.3). Finally, we incorporate a classification network to filter out matches with low confidence to refine the initial matches (Sec.2.3).

2.2.1 Local Feature Extraction

To extract points-wise features from both the 2D keypoint set $\mathbf{P}$ and the 3D point cloud $\mathbf{Q}$, we consider the inputs as bearing vectors equipped with color information: $\mathcal{P} = \{\mathbf{b_p}, \mathbf{c_p}\}$ and $\mathcal{Q} = \{\mathbf{b_q}, \mathbf{c_q}\}$. Two ResNet-style point encoders [4, 17], denoted as $\mathcal{F}_b$ and $\mathcal{F}_c$, are applied to extract position and color embeddings separately. We then obtain the local point features, $\mathbf{f_p}$ and $\mathbf{f_q}$, as follows:

$$\mathbf{f_p} = \mathcal{F}_b(\mathbf{b_p}) + \mathcal{F}_c(\mathbf{c_p}), \quad \mathbf{f_q} = \mathcal{F}_b(\mathbf{b_q}) + \mathcal{F}_c(\mathbf{c_q}). \quad (3)$$

The resulting point-wise features $\mathbf{f_p}$ and $\mathbf{f_q}$ are vectors with dimensions $\mathbb{R}^{N \times d}$ and $\mathbb{R}^{M \times d}$ respectively, where N and M represent the number of keypoints in $\mathbf{P}$ and $\mathbf{Q}$, and d denotes the dimensionality of the encoded features, e.g., $d = 128$.

2.2.2 Global Geometric Guidance

Global context guidance has demonstrated its effectiveness in various computer vision tasks [22, 33, 46, 52]. Global context helps to differentiate local descriptors from similar

structures or patches, thereby reducing ambiguity. However, most existing methods [33, 46] consider the outputs from different encoding layers as global and local features. This approach is not suitable for our scenario, as our input is sparse points. Downsampling the sparse point cloud results in losing distinctive geometric structures. Hence, we adopt cluster-based geometric encoding to extract global embeddings. As shown in Fig. 3.(a) and (c), the input bearing vectors, both in the image and in the point cloud, are first clustered into X groups. The groups represent distinct clusters, each associated with a cluster center as the global position, denoted by $\hat{\mathbf{b}}_{\mathbf{p},x} \in \mathbb{R}^2, x = 1, ...X$. The corresponding global embedding is obtained as the average of the point embeddings within a cluster as $\hat{\mathbf{f}}_{\mathbf{p},x} = \frac{1}{P'}\sum_{p'=1}^{P'} \mathbf{f}_{\mathbf{p},p'}$, where $\hat{\mathbf{f}}_{\mathbf{p},x} \in \mathbb{R}^d$, P' is the number of points in the x. The same is conducted on the 3D points to obtain $\hat{\mathbf{b}}_{\mathbf{q}}$ and $\hat{\mathbf{f}}_{\mathbf{q}}$.

Global Geometric Graph. To aggregate and extract the geometric relations among the clusters, we propose a novel graph neural network that encodes both distance and angular cues; the basic GNN structure is built upon [18, 53]. In the following, we describe the graph construction for the 2D global points set $\hat{\mathcal{P}} = \{\hat{\mathbf{b}}_{\mathbf{p}}, \hat{\mathbf{f}}_{\mathbf{p}}\}$ and the same goes for $\hat{\mathcal{Q}} = \{\hat{\mathbf{b}}_{\mathbf{q}}, \hat{\mathbf{f}}_{\mathbf{q}}\}$. Each cluster center point $\hat{\mathbf{b}}_{\mathbf{p},x}$ is connected to its k-NN neighbours ($k \leq K$) in the coordinate space, and $\xi_{\mathbf{p},(x,y)}$ is the edge between center points $\hat{\mathbf{b}}_{\mathbf{p},x}$ and $\hat{\mathbf{b}}_{\mathbf{p},y}$. We update the feature $\hat{\mathbf{f}}_{\mathbf{p},x}$ using the following equation:

$$^{(t+1)}\hat{\mathbf{f}}_{\mathbf{p},x} = \max_{\xi_{\mathbf{p},(x,y)}} \mathcal{H}_{g1}\left(^{(t)}\hat{\mathbf{f}}_{\mathbf{p},x} \oplus \left(^{(t)}\hat{\mathbf{f}}_{\mathbf{p},x} - ^{(t)}\hat{\mathbf{f}}_{\mathbf{p},y}\right)\right), \quad (4)$$
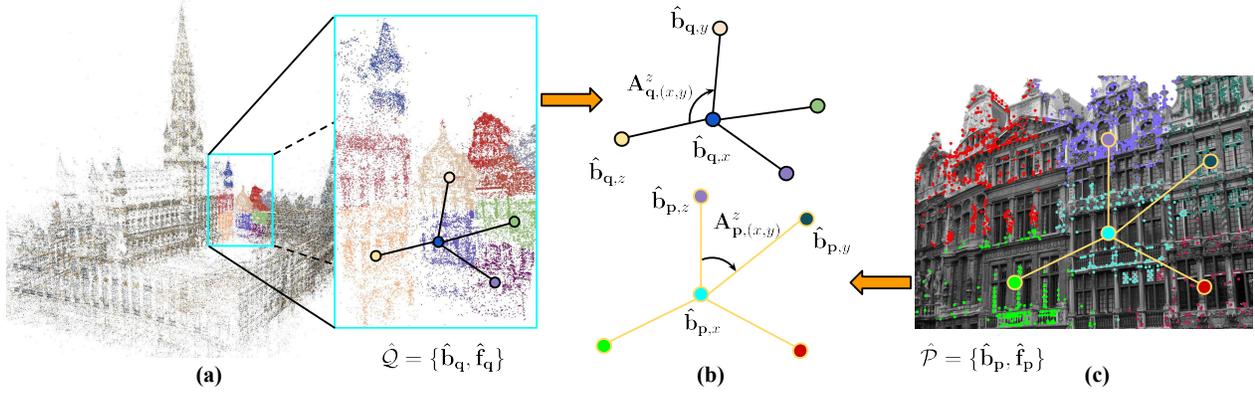
Figure 3. **Cluster-based geometric encoding**. (a) The clusters obtained from bearing vectors $\mathcal{Q}$ of the 3D point cloud are visualized by color. The local graph is created from the neighboring cluster centers. Black 3D points are filtered out from matching. (b) Angular embedding from the global graph to obtain rotation-invariant geometric cues. (c) The clusters obtained from 2D keypoints' bearing vectors $\mathcal{P}$. Similarly, as in 3D, the local graph is created from the neighboring cluster centers.

where the $\oplus$ denotes concatenation and $\mathcal{H}_{g1}(*)$ is the linear projection with instance normalization [48] and a LeakyReLU function [27]. The max operator applies to the k-NN neighbors. The global feature $\hat{\mathbf{f}}_{\mathbf{p},x}$ is updated twice, and calculated as

$$\hat{\mathbf{f}}^g_{\mathbf{p},x} = \mathcal{H}_{g2}(^{(0)}\hat{\mathbf{f}}_{\mathbf{p},x} \oplus^{(1)} \hat{\mathbf{f}}_{\mathbf{p},x} \oplus^{(2)} \hat{\mathbf{f}}_{\mathbf{p},x}). \quad (5)$$

$\mathcal{H}_{g2}$ has a similar structure to $\mathcal{H}_{g1}$, but without shared weights. Besides the distance embedding, inspired by [33], we also adopt the angular embedding to obtain rotation-invariant geometric cues for the global representation. To do so, we define the embedding on cluster triplets as shown in Fig. 3. (b). Given bearing vector $\hat{\mathbf{b}}_{\mathbf{p},x}$ and two of its neighbors $\hat{\mathbf{b}}_{\mathbf{p},y}$ and $\hat{\mathbf{b}}_{\mathbf{p},z}$, the angular embedding of $<\hat{\mathbf{b}}_{\mathbf{p},x}$, $\hat{\mathbf{b}}_{\mathbf{p},y}>$ *w.r.t.* $\hat{\mathbf{b}}_{\mathbf{p},z}$ is defined as follows:

$$\mathbf{A}^z_{x,y} = \text{sine}(\angle(\hat{\mathbf{b}}_{\mathbf{p},z} - \hat{\mathbf{b}}_{\mathbf{p},x}, \quad \hat{\mathbf{b}}_{\mathbf{p},y} - \hat{\mathbf{b}}_{\mathbf{p},x})/\sigma_a), \quad (6)$$

where $\text{sine}(\cdot)$ is a sinusoidal function and σ_a is a controller constant, all k neighbours are considered to obtain the angular embedding $\mathbf{A}_{\mathbf{p}}$. We update the *global geometric embedding* $\hat{\mathbf{f}}^{gg}_{\mathbf{p}}$ as an angular-aware attention mechanism:

$$\hat{\mathbf{f}}^{gg}_{\mathbf{p}} = \text{norm}(\hat{\mathbf{f}}^g_{\mathbf{p}} + \text{Att}(\hat{\mathbf{f}}^g_{\mathbf{p}}, \mathbf{A}_{\mathbf{p}})); \quad \hat{\mathbf{f}}^{gg}_{\mathbf{p}} \in \mathbb{R}^{K \times d}, \quad (7)$$

where

$$\text{Att}(\hat{\mathbf{f}}^g_{\mathbf{p}}, \mathbf{A}_{\mathbf{p}}) =$$
$$(\hat{\mathbf{f}}^g_{\mathbf{p}}\mathbf{W}^{\mathbf{V}}).\frac{(\mathbf{A}_{\mathbf{p}}\mathbf{W}^{\mathbf{A}})(\hat{\mathbf{f}}^g_{\mathbf{p}}\mathbf{W}^{\mathbf{Q}})^{\mathbf{T}} + (\hat{\mathbf{f}}^g_{\mathbf{p}}\mathbf{W}^{\mathbf{Q}})(\hat{\mathbf{f}}^g_{\mathbf{p}}\mathbf{W}^{\mathbf{K}})^{\mathbf{T}}}{\sqrt{dim}}.$$

$\mathbf{W}^{\mathbf{A}}, \mathbf{W}^{\mathbf{Q}}, \mathbf{W}^{\mathbf{K}}, \mathbf{W}^{\mathbf{V}} \in \mathbb{R}^{d \times d}$ are the projection matrices of each item and LayerNorm [2] is applied to Eq. 7. Each local point feature is associated with its corresponding global embedding by $\widetilde{\mathbf{f}}_{\mathbf{p}} = \mathbf{f}_{\mathbf{p}} \oplus \hat{\mathbf{f}}^{gg}_{\mathbf{p}}, \widetilde{\mathbf{f}}_{\mathbf{p}} \in \mathbb{R}^{N \times 2d}$ to obtain $\widetilde{\mathcal{P}} = \{\mathbf{b}_{\mathbf{p}}, \widetilde{\mathbf{f}}_{\mathbf{p}}\}$. The same procedure obtains local and global embedding $\widetilde{\mathcal{Q}} = \{\mathbf{b}_{\mathbf{q}}, \widetilde{\mathbf{f}}_{\mathbf{q}}\}$ for the point cloud $\hat{\mathcal{Q}}$.

2.2.3 Cluster-based Local Matching

After extracting the global geometric embedding, we implement a *cluster-based matching module* to obtain the initial intra-domain 2D-3D matches. This cluster-based GNN [42] has been shown to be more computationally efficient than its complete-graph counterpart [35]. The network considers the local point features from both $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{Q}}$ a complete set, then clusters the feature with strong correlations into the same group and restricts the message passing within each group. In addition to its low computational complexity, we found that cluster GNN can effectively utilize global-to-local geometric cues, as the clustering operation inherits the property of global graph clustering and forces it to distinguish ambiguous local features even with similar global embedding. **Initialization.** As an initialization procedure for the cluster attention module, we run the general self and cross-attention modules proposed in GoMatch [53]. For each local point $\mathbf{b}_{\mathbf{p},n}$, we construct a local graph according to its k' nearest neighbours in the Euclidean space and update the associated feature $\widetilde{\mathbf{f}}_{\mathbf{p},n} \in \mathbb{R}^{2d}$ by Eq. 5. Note that we ignore the angular embedding at this stage due to the unaffordable memory requirements with space complexity $\mathcal{O}(Nk'^2)$, where N is the number of local points. We then use linear attention [20, 46] as a cross-attention mechanism, which allows each point in one modality to interact with all points from another modality. This not only facilitates inter-modality in the feature matching but also reduces the computational complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$.
Cluster-based Attention. After the graph initialization, the features $\widetilde{\mathbf{f}}_{\mathbf{p}}$ and $\widetilde{\mathbf{f}}_{\mathbf{q}}$ coming from the image and the point cloud respectively, are concatenated and processed in a two-level hierarchical clustering attention module. The hierarchical structure is effective in suppressing erroneous groupings. At the first level, we cluster the feature vectors into I coarse

	Methods	ScanNet [9]				MegaDepth [23]			
		Reproj. AUC (%)	Rotation (°)	Translation (m)	P (%) (↑)	Reproj. AUC (%)	Rotation (°)	Translation	P (%) (↑)
		@1 / 5 / 10px (↑)	Quantile @25 / 50 / 75% (↓)			@1 / 5 / 10px (↑)	Quantile @25 / 50 / 75% (↓)		
k=1	Oracle	29.13 / 39.83 / 41.34	0.10 / 0.19 / 0.40	0.01 / 0.01 / 0.03	-	34.59 / 85.02 / 92.02	0.04 / 0.06 / 0.12	0.00 / 0.01 / 0.01	-
	BPnPNet [4]	0.00 / 0.00 / 0.02	99.17 / 128.90 / 154.68	4.35 / 6.82 / 9.86	3.60	0.22 / 0.63 / 0.89	16.13 / 32.01 / 61.58	1.67 / 3.17 / 5.44	2.95
	GoMatch [53]	1.18 / 11.23 / 18.01	2.69 / 12.78 / 36.50	0.19 / 0.91 / 2.63	13.18	5.67 / 22.43 / 28.01	0.60 / 10.08 / 34.63	0.06 / 1.06 / 3.73	14.94
	DGC-GNN	**2.73 / 21.88 / 32.23**	**0.94 / 3.17 / 20.14**	**0.06 / 0.23 / 1.40**	**14.86**	**10.20 / 37.64 / 44.04**	**0.15 / 1.53 / 27.93**	**0.01 / 0.15 / 3.00**	**19.00**
k=10	BPnPNet [4]	0.00 / 0.00 / 0.03	104.68 / 135.94 / 160.54	4.67 / 7.30 / 10.92	0.84	0.36 / 0.72 / 0.97	16.63 / 34.69 / 67.77	1.64 / 3.30 / 5.97	0.74
	GoMatch [53]	0.91 / 18.98 / 31.12	1.18 / 4.94 / 28.97	0.08 / 0.35 / 2.08	4.25	8.90 / 35.67 / 44.99	0.18 / 1.29 / 16.65	0.02 / 0.12 / 1.92	8.76
	DGC-GNN	**1.76 / 31.74 / 48.11**	**0.67 / 1.49 / 7.62**	**0.04 / 0.11 / 0.53**	**6.42**	**15.30 / 51.70 / 60.01**	**0.07 / 0.26 / 5.41**	**0.01 / 0.02 / 0.57**	**13.36**

Table 1. **2D-3D Matching.** We present AUC scores for reprojection errors threshold at 1, 5, and 10 pixels; rotation and translation error quantiles at 25, 50, and 75%; and matching precision. Parameter k is the number of images retrieved from the database to narrow down the search space. The best results are bold. DGC-GNN nearly doubles the AUC scores of GoMatch and reduces the pose errors to their $\approx 33\%$.

groups. In the second level, each coarse group is divided into several small groups. The local point information exchange is conducted at each level and only within each group to obtain more representative features. After the sparse clustering, each feature vector is transformed back to its original position and then split again into $\widetilde{\mathbf{f}}'_{\mathbf{p}}$ and $\widetilde{\mathbf{f}}'_{\mathbf{q}}$ to obtain the keypoints both in the 2D and 3D spaces.

Optimal Transport. We calculate the cost matrix $\mathcal{M} \in \mathbb{R}^{N \times M}$ between the two transformed feature sets using the L_2 distance between pairs of features. Thus, $\mathcal{M}(n, m) = ||\widetilde{\mathbf{f}}'_{\mathbf{p}, n} - \widetilde{\mathbf{f}}'_{\mathbf{q}, m}||_2$. Following [35], the cost matrix $\mathcal{M}$ is extended to $\bar{\mathcal{M}}$ by adding an additional row and column as dustbins for unmatched points. We then iteratively optimize $\bar{\mathcal{M}}$ running the Sinkhorn algorithm [8, 44] in a declarative layer to obtain the score matrix $\bar{\mathcal{S}}$. Finally, $\bar{\mathcal{S}}$ is converted to $\mathcal{S} \in \mathbb{R}^{N \times M}$ by dropping the dustbins. The initial 2D-3D match candidates are acquired by mutual top-1 search, thus

$$\mathcal{M}_{init} = \{(\widetilde{n}, \widetilde{m}) \mid \forall (\widetilde{n}, \widetilde{m}) \in \mathrm{MNN}(\mathcal{S})\}, \quad (8)$$

where MNN is the mutual nearest neighbors operator. Set $\mathcal{M}_{init}$ provides initial 2D-3D matches that we further filter in Sec. 2.3 to keep the accurate correspondences only.

2.3. Outlier Rejection

After obtaining the initial matches, outlier pruning runs to remove the incorrect ones. We apply the same outlier rejection network as in GoMatch [53], whose input is the concatenated 2D and 3D keypoint features $\widetilde{\mathbf{f}}'_{\widetilde{n}, \widetilde{m}} = \widetilde{\mathbf{f}}'_{\mathbf{p}, \widetilde{n}} \oplus \widetilde{\mathbf{f}}'_{\mathbf{q}, \widetilde{m}}$ and outputs the matching confidence of each matched pair. The final predicted matches are obtained as follows:

$$\mathcal{M}_{final} = \{(\widetilde{n}', \widetilde{m}') \mid \forall \, \mathrm{cls}(\widetilde{\mathbf{f}}'_{\widetilde{n}, \widetilde{m}} \mid (\widetilde{n}, \widetilde{m}) \in \mathcal{M}_{init}) \geq \theta\}, \quad (9)$$

where θ is the matching confidence threshold.

2.4. Training Loss

We use the same training loss as GoMatch. The loss function $\mathcal{L}$ consists of two terms, the matching loss $\mathcal{L}_{ot}$ and the classification loss $\mathcal{L}_{or}$. The ground truth match set $\mathcal{M}_{gt}$ is estimated by reprojecting the 3D points to the 2D image plane and calculating the pixel distance. We also include point sets $\mathcal{I}$ and $\mathcal{J}$ for the unmatched points in $\mathcal{P}$ and $\mathcal{Q}$, respectively. The matching loss $\mathcal{L}_{ot}$ minimizes the negative log-likelihood of the matching score $\bar{\mathcal{S}}$.

$$\mathcal{L}_{ot} = -\frac{1}{|\mathcal{M}_{gt}| + |\mathcal{I}| + |\mathcal{J}|} \Big(\sum_{(n,m) \in \mathcal{M}_{gt}} \log \bar{\mathcal{S}}_{n,m} + \sum_{i \in \mathcal{I}} \log \bar{\mathcal{S}}_{i,m+1} + \sum_{j \in \mathcal{J}} \log \bar{\mathcal{S}}_{N+1,j} \Big). \quad (10)$$

The classification loss is defined as

$$\mathcal{L}_{or} = -\frac{1}{|\mathcal{M}_{init}|} \sum_{i=1}^{|\mathcal{M}_{init}|} w_i (y_i \log(p_i) + (1 - y_i) \log(1 - p_i)), \quad (11)$$

where w_i is the weight balancing the positive and negative samples, y_i is the ground truth matching label for the i-th correspondences, p_i is the predicted probability of a true match for the i-th correspondences. The total loss is the sum of the two terms as $\mathcal{L} = \mathcal{L}_{ot} + \mathcal{L}_{or}$.

3. Experiments

Training. We train the indoor model of DGC-GNN on the ScanNet [9] dataset and the outdoor model on the MegaDepth [23] dataset. We extract up to 1024 keypoints for each training image by the SIFT detector [26]. Similarly as in GoMatch, we first select a subset of the point cloud by applying image retrieval approaches [1, 47] to obtain potential images observing the same part of the scene as the input one. We randomly sample the retrieval pairs with a visual overlap of more than 35% on MegaDepth and 65% on ScanNet to ensure enough matches on each pair. For the global

geometric embedding, we cluster the 2D/3D bearing vectors into $X = 10$ groups, and each cluster center is connected to its $k = 4$ nearest neighbors to build the global graph. For the local point graph, we connect each point with its 10 nearest neighbors and the cluster-based attentions are performed twice to force the intra-cluster information exchange.

We use Adam optimizer with a learning rate of 1e-3. We train DGC-GNN with one 32GB Telsa V100 GPU. The convergence of the model typically requires 50 epochs.

Datasets. We use ScanNet and MegaDepth for training and 2D-3D matching task evaluation. As a downstream application, we perform visual localization on the 7Scenes [43] and Cambridge Landmarks [21] datasets. MegaDepth is a popular outdoor dataset with 196 scenes captured around the world. The sparse 3D reconstructions are provided by the COLMAP [40] structure-from-motion software. Following [53], we train our outdoor model on 99 scenes and evaluate it with 53 scenes. ScanNet is a large-scale RGB-D indoor dataset comprising 1613 scans with over 2.5 million images. We randomly selected 105 scenes for the training and 30 for the evaluation. Cambridge Landmarks is a middle-scale outdoor dataset consisting of 6 individual scenes. A structure-from-motion algorithm provides the ground truth camera poses. We follow [21, 53] to evaluate our method on four scenes. 7Scenes is a small indoor dataset with RGB-D images and camera poses provided by the depth SLAM system. We evaluate on the standard test sequences.

Evaluation Protocol. For matching on ScanNet and MegaDepth, we follow [53] and report the AUC score calculated from the reprojection errors. To calculate the errors for the 2D-3D matches in $\mathcal{M}_{final}$, we project the 3D points to the image plane using the ground truth and estimated camera poses. Then, we calculate the L_2 distance of the ground truth and estimated reprojected 2D points. We use multiple thresholds, 1, 5, and 10 pixels, to evaluate the AUC scores. The camera translation and rotation error quantiles at 25%, 50%, and 75% are also reported. Moreover, we evaluate the matching quality by calculating the matching precision P, which is the ratio of inlier matches after PnP-RANSAC to the number of final matches $\mathcal{M}_{final}$. For visual localization tasks, we report the median translation (in meters) and rotation (in degrees) camera pose errors.

3.1. 2D-3D Matching

We compare with the two descriptor-free matchers Go-Match [53] and BPnPnet [4]. At inference, we use the 3D points from the top-k retrieved database images to match with the keypoints from query images. Following [53], we report the upper bound of the AUC score using the ground truth matches. We refer to these values as Oracle. We select the GT matches by thresholding the reprojection error based on normalized image coordinates, using a threshold of 0.001, to bypass the influence of camera intrinsics during
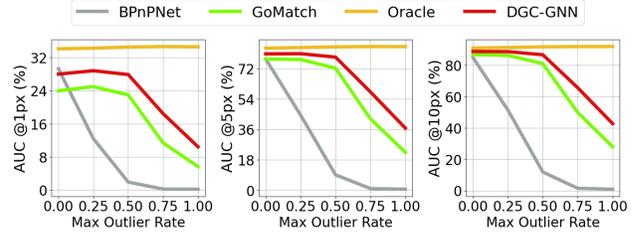

Figure 4. **Outlier Sensitivity**. The AUC scores of BPnPNet [4], GoMatch [53], and the proposed DGC-GNN thresholded at 1, 5, and 10 pixels are plotted as a function of the outlier ratio. Oracle represents the AUC upper bound using ground truth matches.

GT selection. This is in contrast to what is done in [53]. Results on GT selected by a pixel threshold are in the supp. material. We use the official code with the default setting to generate the evaluation dataset on MegaDepth [23] and rerun GoMatch and BPnPNet with the released models. Note that we also tested GoMatch after retraining it on MegaDepth and achieved similar results as with the released model.

Matching Results. The results with $k = 1$ and $k = 10$ are presented in Table 1. Parameter k is the number of retrieved image pairs that are used for evaluation. The proposed method outperforms GoMatch and BPnPNet by a significant margin on both scenes. Specifically, DGC-GNN achieves 10.2 / 37.64 / 44.04% reprojection AUC compared to GoMatch with 5.67 / 22.43 / 28.01% on MegaDepth with $k = 1$. DGC-GNN halves the rotation and translation errors of GoMatch on all thresholds and it obtains better matching quality. Notably, the performance of DGC-GNN with $k = 1$ surpasses that of GoMatch with $k = 10$, indicating the effectiveness of our method even with a single view.

Sensitivity to Outliers. To evaluate the sensitivity to keypoint outliers, we follow the procedure in GoMatch [53]. The outliers are controlled by the outlier ratio, ranging from 0 to 1, calculated as the number of unmatched keypoints divided by the maximum of the numbers of 2D and 3D points. If the outlier ratio is 0, all the input 2D and 3D points are selected from the ground truth matches, and no outliers are included in the matching process. When it is 1, we directly use the keypoints from the query image and 3D points from the top-k retrieved images without any filtering or outlier removal. The results are shown in Fig. 4. Even in the presence of outliers, DGC-GNN outperforms other methods by a large margin. This indicates that our method is more robust to outliers and can handle challenging matching scenarios more effectively than the state-of-the-art.

Ablation Study. We investigate the effectiveness of different components of DGC-GNN on the 2D-3D matching quality on the MegaDepth dataset [23] with $k = 10$. The results are reported in Table 2. We provide results with $k = 1$ in the supp. material. We conduct the ablations by gradually

Methods	G. Emb. Sec. 2.2.2	C. Att. Sec. 2.2.3	Color Sec. 2.2.1	Ang. Sec. 2.2.2	Reproj. AUC (%) @1 / 5 / 10px (↑)	Rotation (°) Quantile@25 / 50 / 75% (↓)	Translation
GoMatch [53]					8.90 / 35.67 / 44.99	0.18 / 1.29 / 16.65	0.02 / 0.12 / 1.92
Variants	✓				10.86 / 41.18 / 50.51	0.13 / 0.76 / 13.47	0.01 / 0.07 / 1.62
	✓	✓			11.64 / 44.46 / 53.99	0.11 / 0.55 / 9.49	0.01 / 0.05 / 1.05
		✓	✓		13.20 / 46.33 / 54.34	0.09 / 0.41 / 9.98	0.01 / 0.03 / 1.19
	✓	✓	✓		14.19 / 48.34 / 56.54	0.08 / 0.34 / 9.23	0.01 / 0.03 / 1.03
DGC-GNN	✓	✓	✓	✓	**15.30 / 51.70 / 60.01**	**0.07 / 0.26 / 5.41**	**0.01 / 0.02 / 0.57**

Table 2. **Ablation Study.** AUC scores thresholded at 1, 5, and 10 pixels; rotation and translation error quantiles at 25, 50, 75% with the proposed components added one by one to the GoMatch pipeline on the MegaDepth dataset.

adding the components: global geometric embedding (G. Emb), cluster attention (C. Att.), Color, and Angular embedding (Ang.) to the original GoMatch pipeline. Incorporating color information into the matching process significantly impacts the performance, resulting in improvements of 2.55 / 3.88 / 2.55% (AUC@1 / 5 / 10px). This demonstrates the importance of color cues for accurate and robust matching. The global-to-local geometric (G. Emb.) and Angular relation embedding (Ang.) substantially improve the matching performance by 1.90 / 5.51 / 5.52% and 1.11 / 3.36 / 3.47%, respectively. It highlights the effectiveness of incorporating global geometric context and local geometric details. The cluster attention mechanism also plays a vital role, improving performance by 0.78 / 3.28 / 3.48%. The best results are obtained when all components are added to the pipeline.

3.2. Visual Localization

Visual localization estimates the 6 degrees-of-freedom camera pose of an input query image *w.r.t* a known map of the scene. One of the most prominent ways of approaching this problem is via establishing 2D-3D correspondences and running robust pose estimation. Following [53], we ran the proposed DGC-GNN to obtain matches. For each query image, we match its keypoints with the 3D points from the top-10 retrieved views to build the 2D-3D correspondences. The camera pose is then estimated by PnP-RANSAC [15, 16]. We use two standard datasets, 7Scenes [43] and Cambridge Landmarks [21]. For 7Scenes, we extract the keypoints with the SIFT detectors, and the top 10 pairs are retrieved using DenseVLAD [47]. For Cambridge Landmarks, the keypoints are extracted by SuperPoint [10] to ensure consistency with the SuperPoint-based structure-from-motion model. The top 10 pairs are provided by NetVLAD [1].

3.2.1 Results

In Table 3, we present the 3D model maintenance costs, privacy, storage requirements, and camera pose median er-

rors (cm, °) of standard descriptor-based localization techniques and descriptor-free methods. DGC-GNN consistently outperforms GoMatch on *all* scenes by a significant margin. On Cambridge Landmarks, the average median error of DGC-GNN is 54 cm / 2.23°, while GoMatch leads to 173 cm / 5.87° error. On 7Scenes, the average error of DGC-GNN is 15 cm / 4.47°, and that of GoMatch is 22 cm / 5.77°. DGC-GNN requires a similar amount of memory to other descriptor-free methods. Also, it inherits their privacy-preserving properties due to not requiring visual descriptors.

The trade-off between descriptor-based (DB) and descriptor-free (DF) algorithms is visible from the table. While descriptor-based ones lead to the best accuracy overall, they require excessive memory and descriptor maintenance and are susceptible to privacy attacks. Although the model compression method, HybridSC [5], shows effectiveness in storage saving, it achieves similar performance compared to DGC-GNN on the Cambridge Landmarks dataset while still requiring descriptor maintenance. End-to-end methods (E2E) overcome these problems and achieve accurate results. However, their main limitation is that such approaches must be trained independently on each scene. The proposed DGC-GNN only needs to be trained once, making it more efficient and convenient to use as an off-the-shelf tool.

3.2.2 Generalizability

Similar to [53], we discuss the generalizability of our DGC-GNN model on the visual localization task across different training and evaluation scenes. Specifically, we investigate the performance of our model when trained on MegaDepth [23] and ScanNet [9] and evaluated on the 7Scenes dataset [43]. We also explore the impact of using different keypoint detectors, namely SIFT [26] and SuperPoint [10], during the evaluation.

These experiments are summarized in Table 4, providing an overview of the performance of DGC-GNN under different training and evaluation conditions. While the best overall performance is achieved by training on the MegaDepth

Methods	No Desc. Maint.	Privacy	Cambridge-Landmarks [21] (cm, °)				MB used	7Scenes [43] (cm, °)							MB used
			King's	Hospital	Shop	St. Mary's		Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	
E2E MS-Trans. [41]	✓	✓	83 / 1.47	181 / 2.39	86 / 3.07	162 / 3.99	71	11 / 4.66	24 / 9.60	14 / 12.19	17 / 5.66	18 / 4.44	17 / 5.94	26 / 8.45	71
DSAC* [3]	✓	✓	**15 / 0.30**	21 / 0.40	**5 / 0.30**	13 / 0.40	112	**2 / 1.10**	**2 / 1.24**	**1 / 1.82**	3 / 1.15	4 / 1.34	4 / 1.68	**3 / 1.16**	196
HSCNet [22]	✓	✓	18 / 0.30	**19 / 0.30**	6 / 0.30	**9 / 0.30**	592	**2 / 0.70**	**2 / 0.90**	**1 / 0.90**	3 / 0.80	4 / 1.00	4 / 1.20	**3 / 0.80**	1036
DB HybridSC [5]	✗	–	81 / 0.59	75 / 1.01	19 / 0.54	50 / 0.49	3	-	-	-	-	-	-	-	-
AS [38]	✗	✗	13 / 0.22	20 / 0.36	**4 / 0.21**	8 / 0.25	813	3 / 0.87	**2 / 1.01**	**1 / 0.82**	4 / 1.15	7 / 1.69	5 / 1.72	**4 / 1.01**	-
SP [10]+SG [35]	✗	✗	**12 / 0.20**	**15 / 0.30**	**4 / 0.20**	**7 / 0.21**	3215	**2 / 0.85**	**2 / 0.94**	**1 / 0.75**	**3 / 0.92**	5 / 1.30	**4 / 1.40**	5 / 1.47	22977
DF GoMatch [53]	✓	✓	25 / 0.64	283 / 8.14	48 / 4.77	335 / 9.94	48	4 / 1.65	13 / 3.86	9 / 5.17	11 / 2.48	16 / 3.32	13 / 2.84	89 / 21.12	302
DGC-GNN	✓	✓	**18 / 0.47**	**75 / 2.83**	**15 / 1.57**	**106 / 4.03**	69	**3 / 1.43**	**5 / 1.77**	**4 / 2.95**	**6 / 1.61**	**8 / 1.93**	**8 / 2.09**	**71 / 19.5**	355

Table 3. **Visual Localization.** We report the median pose errors (cm, °) and storage requirements (MB) on the scenes of the 7Scenes [43] and Cambridge-Landmarks [21] datasets. Three groups of methods are shown: end-to-end (E2E), descriptor-based (DB), and descriptor-free (DF). We do not show BPnPNet as it fails on most scenes. The best results are shown in bold in each group.

Scenes	Trained on MegaDepth [23] (SIFT)				Trained on ScanNet [9] (SIFT)	
	GoMatch (SIFT)	GoMatch (SP)	DGC-GNN (SIFT)	DGC-GNN (SP)	DGC-GNN (SIFT)	DGC-GNN (SP)
Chess	4 / 1.65	4 / 1.56	**3 / 1.41**	3 / 1.46	**3 / 1.43**	4 / 1.51
Fire	13 / 3.86	12 / 3.71	**5 / 1.81**	7 / 2.30	**5 / 1.77**	6 / 2.03
Heads	9 / 5.17	5 / 3.43	5 / 3.13	**4 / 2.78**	4 / 2.95	4 / 3.02
Office	11 / 2.48	7 / 1.76	7 / 1.66	7 / 1.66	**6 / 1.61**	7 / 1.66
Pumpkin	16 / 3.32	28 / 5.65	**8 / 2.03**	12 / 2.75	**8 / 1.93**	10 / 2.38
Redkitchen	13 / 2.84	14 / 3.03	**8 / 2.14**	10 / 2.36	**8 / 2.09**	9 / 2.28
Stairs	89 / 21.12	58 / 13.12	83 / 21.53	**55 / 13.05**	71 / 19.50	58 / 14.32
All	22 / 5.78	18 / 4.61	17 / 4.82	**14 / 3.77**	15 / 4.47	**14 / 3.89**

Table 4. **Model Generalizability on Visual Localization Task.** We report the translation and rotation median error (cm / °) on 7Scenes dataset [43]. We evaluate the models trained on MegaDepth [23] and ScanNet [9] datasets. The best performance is bold.

dataset with SIFT features, the results are similar in both training scenarios, showcasing that the proposed method generalizes well to unseen data. While we train on SIFT features, the best results are achieved by using SuperPoint features at inference time. This demonstrates that DGC-GNN is insensitive to the features used and can be utilized off-the-shelf even without retraining to our specific scenario.

4. Conclusion

In conclusion, this paper introduces DGC-GNN, a novel graph-based pipeline for visual descriptor-free 2D-3D matching that effectively leverages geometric and color cues in a global-to-local manner. Our global-to-local procedure encodes both Euclidean and angular relations at a coarse level, forming a geometric embedding to guide the local point matching. By employing a cluster-based transformer, we enable efficient information passing within local clusters, ultimately leading to significant improvements in the number of correct matches and the accuracy of pose estimation. Compared to the state-of-the-art descriptor-free matcher Go-Match [53], the proposed DGC-GNN demonstrates a sub-

stantial improvement, doubling the accuracy on real-world and large-scale datasets. Furthermore, it results in significantly increased localization accuracy. These advancements contribute to reducing the gap between descriptor-based and descriptor-free methods while addressing the limitations of descriptor-based ones, such as memory footprint, maintenance costs, and susceptibility to privacy attacks.

Limitations. The primary limitation of our proposed DGC-GNN method lies in its performance being inferior to traditional descriptor-based algorithms. The performance difference can be attributed to the insufficiency of unique 3D structures in the geometry, which hinders the ability of the algorithm to identify distinct matches in real-world scenarios. Although DGC-GNN demonstrates a notable improvement over existing descriptor-free approaches, there remains a performance gap to overcome in order to achieve results on par with or superior to those of descriptor-based methods.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. 1, 5, 7

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[3] Eric Brachmann and Carsten Rother. Visual camera relocalization from RGB and RGB-D images using DSAC. *TPAMI*, 2021. 2, 8

[4] Dylan Campbell, Liu Liu, and Stephen Gould. Solving the blind perspective-n-point problem end-to-end with robust differentiable geometric optimization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, page preprint. Springer, 2020. 2, 3, 5, 6

[5] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid scene compression for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7653–7662, 2019. 7, 8

[6] Kunal Chelani, Fredrik Kahl, and Torsten Sattler. How privacy-preserving are line clouds? recovering scene details from 3d lines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15668–15678, 2021. 1

[7] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 864–872, 2015. 1

[8] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013. 5

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 5, 7, 8

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, pages 224–236, 2018. 1, 7, 8

[11] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29, 2016. 1

[12] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4829–4837, 2016. 1

[13] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8092–8101, 2019. 1

[14] Ethan Eade and Tom Drummond. Scalable monocular slam. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 469–476. IEEE, 2006. 1

[15] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 7

[16] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003. 7

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[18] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 4267–4276, 2021. 3

[19] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *in IEEE Conference on Computer Vision and Pattern Recognition*, pages 2599–2606. IEEE, 2009. 1

[20] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 4

[21] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 2, 6, 7, 8

[22] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11983–11992, 2020. 2, 3, 8

[23] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 2, 5, 6, 7, 8

[24] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5987–5997, 2021. 1

[25] Liu Liu, Dylan Campbell, Hongdong Li, Dingfu Zhou, Xibin Song, and Ruigang Yang. Learning 2d-3d correspondences to solve the blind perspective-n-point problem. *arXiv preprint arXiv:2003.06752*, 2020. 2

[26] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1, 5, 7

[27] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the International Conference of Machine Learning*, page 3, 2013. 4

[28] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 1

[29] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 1

[30] Tony Ng, Hyo Jin Kim, Vincent T Lee, Daniel DeTone, Tsun-Yi Yang, Tianwei Shen, Eddy Ilg, Vassileios Balntas, Krystian Mikolajczyk, and Chris Sweeney. Ninjadesc: content-concealing visual descriptors via adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12797–12807, 2022. 2

[31] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 2161–2168. Ieee, 2006. 1

[32] Linfei Pan, Johannes L Schönberger, Viktor Larsson, and Marc Pollefeys. Privacy preserving localization via coordinate permutations. In *Proceedings of the International Conference on Computer Vision*, pages 18174–18183, 2023. 1

[33] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11143–11152, 2022. 3, 4

[34] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019. 1, 2

[35] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4938–4947, 2020. 1, 2, 4, 5, 8

[36] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *International Conference on Computer Vision*, pages 667–674. IEEE, 2011. 1

[37] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *European conference on computer vision*, pages 752–765. Springer, 2012. 1

[38] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1744–1756, 2016. 1, 2, 8

[39] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, 2018. 1

[40] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 1, 6

[41] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2733–2742, 2021. 8

[42] Yan Shi, Jun-Xiong Cai, Yoli Shavit, Tai-Jiang Mu, Wensen Feng, and Kai Zhang. Clustergnn: Cluster-based coarse-to-fine graph neural network for efficient feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12517–12526, 2022. 1, 2, 4

[43] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013. 2, 6, 7, 8

[44] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967. 5

[45] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph*, pages 835–846. 2006. 1

[46] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 3, 4

[47] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1808–1817, 2015. 5, 7

[48] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[50] Shuzhe Wang, Zakaria Laskar, Iaroslav Melekhov, Xiaotian Li, and Juho Kannala. Continual learning for image-based camera localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3252–3262, 2021. 1

[51] Luwei Yang, Rakesh Shrestha, Wenbo Li, Shuaicheng Liu, Guofeng Zhang, Zhaopeng Cui, and Ping Tan. Scenesqueezer: Learning to compress scene for camera relocalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8259–8268, 2022. 1

[52] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *Advances in Neural Information Processing Systems*, 34:23872–23884, 2021. 3

[53] Qunjie Zhou, Sérgio Agostinho, Aljoša Ošep, and Laura Leal-Taixé. Is geometry enough for matching in visual localization? In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 407–425. Springer, 2022. 1, 2, 3, 4, 5, 6, 7, 8