U-Match: Two-view Correspondence Learning with Hierarchy-aware Local Context Aggregation

Zizhuo Li , **Shihua Zhang** and **Jiayi Ma***

Electronic Information School, Wuhan University, Wuhan 430072, China

zizhuo_li@whu.edu.cn, suhzhang001@gmail.com, jyma2010@gmail.com

Abstract

Local context capturing has become the core factor for achieving leading performance in two-view correspondence learning. Recent advances have devised various local context extractors whereas typically adopting *explicit* neighborhood relation modeling that is restricted and inflexible. To address this issue, we introduce U-Match, an attentional graph neural network that has the flexibility to enable *implicit* local context awareness at multiple levels. Specifically, a *hierarchy-aware graph representation* (HAGR) module is designed and fleshed out by local context pooling and unpooling operations. The former encodes local context by adaptively sampling a set of nodes to form a coarse-grained graph, while the latter decodes local context by recovering the coarsened graph back to its original size. Moreover, an *orthogonal fusion* module is proposed for the collaborative use of HAGR module, which integrates complementary local and global information into compact feature representations without redundancy. Extensive experiments on different visual tasks prove that our method significantly surpasses the state-of-the-arts. In particular, U-Match attains an AUC at 5 degree threshold of 60.53% on the challenging YFCC100M dataset without RANSAC, outperforming the strongest prior model by 8.61 absolute percentage points. Our code is publicly available at https://github.com/ZizhuoLi/U-Match.

1 Introduction

Identifying reliable correspondences between two-view images and retrieving the camera motion encoded by the essential matrix, continues to be an important and general problem in computer vision, with applications to Simultaneous Localization and Mapping (SLAM) [Ma *et al.*, 2021], Structure-from-Motion (SfM) [Schonberger and Frahm, 2016], visual localization [Sattler *et al.*, 2018], image registration and fusion [Tang *et al.*, 2022]. Given a pair of images, the most standard pipeline to address the correspondence learning task follows a three-step strategy, *i.e.*, feature detection, feature description, and feature matching. Specifically, keypoints are first detected from each image and then patches centered at these ones are used to generate corresponding visual descriptors by means of off-the-shelf detector-descriptors [Lowe, 2004; DeTone *et al.*, 2018]. Finally, a Nearest Neighbor (NN) matcher is typically applied to establish point-to-point correspondences between images. Nevertheless, the generated set of putative matches is inevitably largely dominated by false ones (*i.e.*, outliers), as a result of the ambiguity of visual descriptors, especially in the case of extreme situations, including poor texture, repetitive elements, viewpoint change, illumination variation, and motion blur. Therefore, many recent researches have been dedicated to designing accurate and robust correspondence learning methods, with the purpose of rejecting spurious matches while maintaining as many true ones (*i.e.*, inliers) as possible.

As the most prevalent paradigm among hand-engineered techniques for correspondence learning, RANSAC [Fischler and Bolles, 1981] and its variants [Raguram *et al.*, 2012; Barath *et al.*, 2020] seek inliers by finding the largest subset conforming to a task-specific geometric model such as affine, homography, or epipolar geometry. However, the theoretical running time of such methods increases exponentially with respect to the outlier rate. Thus, another line of research explores smoothness constraints to handle this drawback [Ma *et al.*, 2014; Bian *et al.*, 2017; Ma *et al.*, 2019]. Nevertheless, these constraints are tough to uphold in the case of large outlier rates, resulting in dramatically degraded performance.

Due to the astonishing performance achieved by the recent developments in deep learning, end-to-end trainable Multi-Layer Perceptrons (MLPs) have been widely applied for two-view correspondence learning to address the irregular and unordered characteristics of sparse matches. Motivated by PointNet [Qi *et al.*, 2017], the pioneering work PointCN [Yi *et al.*, 2018] adopts a PointNet-like architecture to process each correspondence individually and predict the inlier likelihood scores of correspondences. Wherein, a non-parametric operation called Context Normalization is introduced to capture global context, realized by just simply normalizing the distribution of the feature maps with their mean and variance. To mitigating the impact of outliers on Context Normalization, PointACN [Sun *et al.*, 2020] proposes to weigh the importance of each feature map during the normalization

*Corresponding author

step. However, both of them neglect the underlying local geometric relations among tentative correspondences and impeding the correspondence learning performance. To mitigate this downside, a plethora of follow-ups have made notable contributions to the design of delicate local geometric extractors. Particularly, OANet [Zhang *et al.*, 2019] designs a clustering module, which maps correspondences to a set of clusters in a soft assignment manner, for local context capturing. CLNet [Zhao *et al.*, 2021] stacks multiple local-to-global consensus learning layers to progressively reject mismatches, where an annular convolutional operation is introduced to aggregate local features. LMCNet [Liu *et al.*, 2021] presents coherence residual and local coherence layers to model motion coherence property for correspondence learning. MS^{2}DG-Net [Dai *et al.*, 2022] introduces dynamic sparse semantic graphs to capture local topology among correspondences. While these methods demonstrate attractive performance in some generic scenarios, most of them typically extract local context by means of explicit neighborhood relation modeling (*e.g.*, k-nearest neighbors), which is limited and inflexible for the correspondence learning task due to two reasons: 1) putative matches are typically non-uniformly distributed across the image domain, and 2) the number of inliers varies with different scenarios, resulting in that locally consistent geometric cues cannot be fully captured.

In this paper, we propose a simple yet surprisingly effective *Attentional Graph Neural Network* dubbed U-Match to address the above challenges. Specifically, we present local context pooling (LCPool) and unpooling (LCUnpool) operations to build a UNet-like multi-level architecture (thereby we name our method U-Match), *i.e.*, *hierarchy-aware graph representation* (HAGR) module, flexibly encoding and decoding high-level features for implicit local context aggregation. The LCPool layer adaptively selects several nodes to form a coarse-grained graph for local context encoding. The LCUnpool layer restores the coarse-grained graph into its original size for local context decoding. Considering that global context is also conducive to correspondence learning and works collaboratively with local context, we further introduce an *orthogonal fusion* (OF) module to combine complementary local and global context, thus generating redundancy-free compact feature representations. Comprehensive experiments on real-world tasks such as relative pose estimation, homography estimation, and visual localization reveal that our method outperforms long lines of prior work.

To sum up, our main contributions are as follows:

- Instead of explicitly capturing local context as most existing studies have done, we introduce an HAGR module, which has the flexibility to enable implicit local context awareness at multiple levels, thus fully exploring underlying local geometric cues.
- We propose an OF module to integrate complementary local and global context without redundancy.
- We design an attentional graph neural network for two-view correspondence learning, which can remove spurious matches from candidate ones effectively.
- We achieve state-of-the-art results on relative pose estimation, homography estimation, and visual localization.

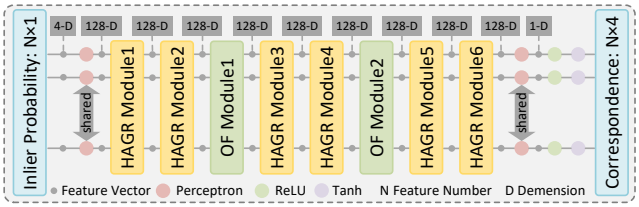

Figure 1: Network architecture of U-Match, which takes the putative correspondences as input and outputs the inlier probability of each correspondence.

2 U-Match

We propose a top-performing graph neural network termed as U-Match for two-view correspondence learning. As illustrated in Fig. 1, our network contains: 1) HAGR module that enables implicit local context awareness at multiple levels, and 2) OF module that realizes the complementary of local and global context without redundancy. In the following parts, the general formulation of our problem will be introduced first, followed by detailed description of each module.

2.1 Problem Formulation

Give an image pair $(\mathbf{I}, \mathbf{I}')$ depicting the same visual content, the objective of our task is to seek accurate and geometrically consistent correspondences and utilize them to recover the relative camera pose accordingly. To this end, any off-the-shelf detector-descriptors can be first used to detect 2D keypoints and generate their corresponding visual descriptions, either handcrafted methods (*e.g.*, SIFT [Lowe, 2004]) or learning-based ones (*e.g.*, SuperPoint [DeTone *et al.*, 2018]). Then, a group of N initial correspondences $\mathbf{C}$ can be generated via an NN matcher:

$$\mathbf{C} = [\mathbf{c}_1; \mathbf{c}_2; \cdots; \mathbf{c}_N] \in \mathcal{R}^{N\times 4}, \tag{1}$$

where $\mathbf{c}_i = (x_i, y_i, x_i', y_i')$ is a correspondence, (x_i, y_i) and (x_i', y_i') are coordinates of keypoints in the image pair, both of which are normalized to $[-1, 1]$ with the camera intrinsics.

Following the *de facto* standard [Yi *et al.*, 2018; Zhang *et al.*, 2019], we formulate the two-view correspondence learning task as an inlier/outlier classification problem and an essential matrix regression problem. An overview of U-Match's workflow is illustrated in Fig. 1. Given the initial correspondences $\mathbf{C}$ as input, a shared-weight MLP first processes them to generate C-dimension feature vectors (here we choose $C = 128$), which are subsequently fed into six HAGR modules and two OF modules, respectively. Then, the features output by the last HAGR module are possessed with an inlier predictor (*i.e.*, a shared-weight MLP followed by tanh and ReLU activation functions), to output the probability set $\mathbf{w} = [w_1, w_2, \cdots, w_N]^\top \in \mathcal{R}^{N\times 1}$, where $w_i \in [0, 1)$ indicates the inlier probability of correspondence $\mathbf{c}_i$, *i.e.*, $w_i = 0$ represents an outlier. Lastly, the weighted eight-point algorithm [Yi *et al.*, 2018] is leveraged to directly regress the essential matrix $\widehat{\mathbf{E}}$ grounded on the probability set $\mathbf{w}$. The whole procedure described above can be summarized as:

$$\begin{aligned} \mathbf{z} &= f_\phi(\mathbf{C}), \\ \mathbf{w} &= \sigma(\mathbf{z}), \\ \widehat{\mathbf{E}} &= g(\mathbf{w}, \mathbf{C}), \end{aligned} \tag{2}$$

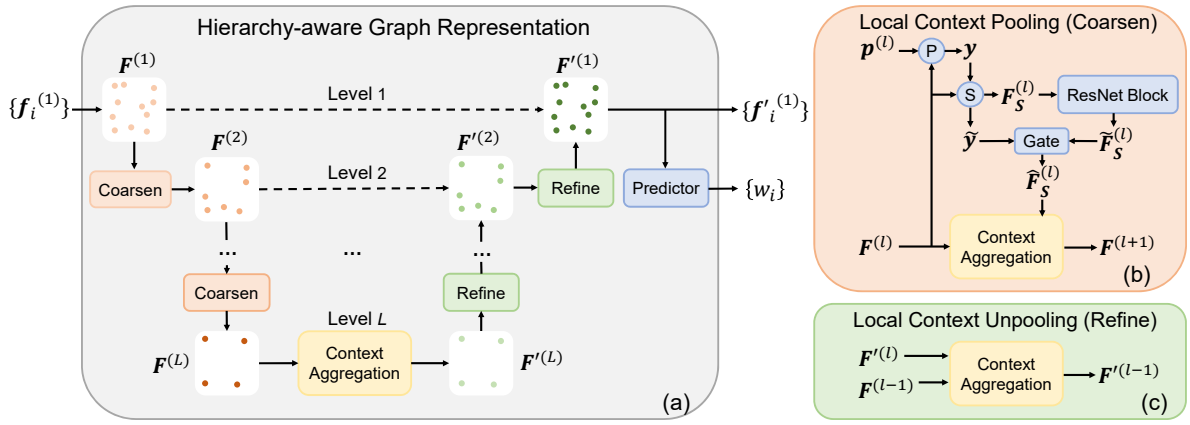

Figure 2: Hierarchy-aware graph representation. (a) Overview of the proposed HAGR module. (b) Workflow of the local context pooling layer. "P" indicates the projection stage and "S" represents the node sampling stage. (c) Workflow of the local context unpooling layer.

where $\mathbf{z}$ denotes the logit values used for classification, $f_\phi(\cdot)$ represents our U-Match with parameters ϕ, $\sigma(\cdot)$ consists of tanh and ReLU activation functions, and $g(\cdot,\cdot)$ denotes the weighted eight-point algorithm that is more robust to outliers than the traditional one, as it has taken into account the inlier confidence of each correspondence, to alleviate the adverse effect of outliers on the regression process.

Next, we discuss the proposed HAGR and OF modules.

2.2 Hierarchy-aware Graph Representation

Due to the variety of data, explicitly aggregating local context has a deficiency in modeling underlying complex local context. Therefore, we propose the HAGR module, to achieve implicit multi-level local context awareness flexibly in an encoder-decoder manner, as shown in Fig. 2(a). The encoder is built by stacking several encoding layers (*i.e.*, LCPool layers), each of which coarsens the graph to encode higher-order local context, as shown in Fig. 2(b). In the decoder part, we stack the same number of decoding layers (*i.e.*, LCUnPool layers), each of which refines the graph into a higher resolution structure to decode fine-grained local context, as shown in Fig. 2(c). Additionally, there are skip-connections between corresponding levels of LCPool and LCUnPool layers, used to conduct encoder-decoder communication.

Local Context Pooling Layer. Given the node features $\mathbf{F}^{(\ell)} = \{\mathbf{f}_i^{(\ell)}\}_{i=1}^N \in \mathcal{R}^{N\times C}$ at level ℓ input to the LCPool layer, we coarsen the graph with the desire to preserve nodes that can characterize the graph best. For this purpose, we introduce a trainable projection vector $\mathbf{p}^{(\ell)} \in \mathcal{R}^C$ to measure the importance of each node to the graph and a sampling ratio $\lambda^{(\ell)} \in (0,1]$ to control the size of the new coarse-grained graph. Specifically, we first calculate the scalar projection value of each node onto $\mathbf{p}^{(\ell)}$ and then adaptively sample a subset of nodes $\mathbf{F}_\mathbf{S}^{(\ell)}$ as follows:

$$\begin{aligned} \mathbf{y} &= \frac{\mathbf{F}^{(\ell)}\cdot\mathbf{p}^{(\ell)}}{\|\mathbf{p}^{(\ell)}\|^2} \in \mathcal{R}^N, \\ \mathcal{I} &= \text{top-rank}(\mathbf{y}, k^{(\ell)}) \in \mathcal{R}^{k^{(\ell)}}, \\ \mathbf{F}_\mathbf{S}^{(\ell)} &= \mathbf{F}^{(\ell)}(\mathcal{I},:) \in \mathcal{R}^{k^{(\ell)}\times C}, \end{aligned} \tag{3}$$

where $\|\cdot\|$ denotes L2-norm, top-rank$(\mathbf{y}, k^{(\ell)})$ is a function that returns the indices $\mathcal{I}$ of the top-k values in the projection score set $\mathbf{y}$, and $k^{(\ell)} = \text{round}(\lambda^{(\ell)}N)$ indicates the number of sampled nodes at level ℓ.

After that, these selected nodes $\mathbf{F}_\mathbf{S}^{(\ell)}$ retrieves information from full nodes $\mathbf{F}^{(\ell)}$ to achieve local context encoding:

$$\begin{aligned} \widetilde{\mathbf{F}}_\mathbf{S}^{(\ell)} &= \mathcal{P}(\mathbf{F}_\mathbf{S}^{(\ell)}), \\ \tilde{\mathbf{y}} &= \text{sigmoid}(\mathbf{y}(\mathcal{I})), \\ \hat{\mathbf{F}}_\mathbf{S}^{(\ell)} &= \widetilde{\mathbf{F}}_\mathbf{S}^{(\ell)} \odot (\tilde{\mathbf{y}}\mathbf{1}_C^\top), \\ \mathbf{F}^{(\ell+1)} &= \mathbf{Cgg}(\hat{\mathbf{F}}_\mathbf{S}^{(\ell)}, \mathbf{F}^{(\ell)}), \end{aligned} \tag{4}$$

where $\mathcal{P}(\cdot)$ denotes ResNet block [Zhao *et al.*, 2021] used for pre-processing, $\tilde{\mathbf{y}}$ is the gate vector that not only controls information flow but also makes the projection vector $\mathbf{p}^{(\ell)}$ trainable by back-propagation, $\mathbf{1}_C \in \mathcal{R}^C$ is a vector with all elements being 1, $\odot$ represents Hadamard product, and $\mathbf{Cgg}(\cdot,\cdot)$ denotes the context aggregation layer in light of an attention mechanism [Vaswani *et al.*, 2017]. Concretely, in a C-dimension feature space, there are M vectors, *i.e.*, $\mathbf{X} \in \mathcal{R}^{M\times C}$, to be updated, and N vectors, *i.e.*, $\mathbf{Y} \in \mathcal{R}^{N\times C}$, to be attended to. The attentional aggregation $\mathbf{Cgg}(\cdot,\cdot)$ can be described as:

$$\begin{aligned} \Delta &= \text{Softmax}(\mathbf{Q}\mathbf{K}^\top)\mathbf{V}, \\ \mathbf{Cgg}(\mathbf{X},\mathbf{Y}) &= \mathbf{X} + \text{MLP}(\mathbf{X}||\Delta), \end{aligned} \tag{5}$$

where $\mathbf{Q}$ is linear projection of $\mathbf{X}$, $\mathbf{K}$ and $\mathbf{V}$ are linear projection of $\mathbf{Y}$. By attentional aggregation, each element in $\mathbf{X}$ can retrieve and aggregate context from elements in $\mathbf{Y}$.

By performing the procedures described above, the output $\mathbf{F}^{(\ell+1)}$ have encoded hierarchical local context from $\mathbf{F}^{(\ell)}$. Notably, for better local context decoding, before feeding $\mathbf{F}^{(\ell+1)}$ into LCUnPool layers, we conduct an intra-graph communication at the coarsest level with a context aggregation layer:

$$\mathbf{F}^{'(L)} = \mathbf{Cgg}(\mathbf{F}^{(L)}, \mathbf{F}^{(L)}), \tag{6}$$

where L is the number of levels and $\mathbf{F}^{'(L)}$ denotes the updated feature nodes input to subsequent LCUnPool layers.

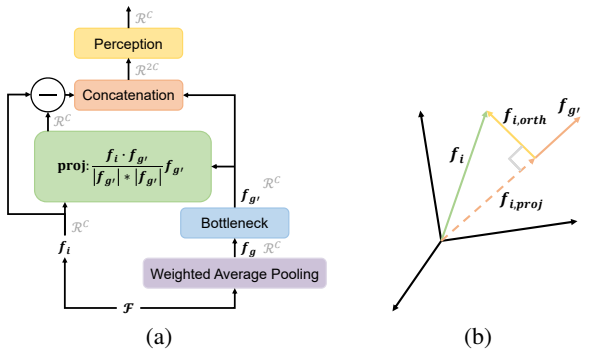

Figure 3: Orthogonal fusion module. (a) Workflow of the *orthogonal fusion* module. (b) Schema of a feature projected onto the global feature and the component orthogonal to the global feature.

Local Context Unpooling Layer. After the progressive local context encoding with LCPool layers and message exchange between coarsest-level node features via the context aggregation layer, we adopt a straightforward way to implement the LCUnPool layer at level ℓ as follows:

$$\mathbf{F}^{'(\ell-1)} = \mathbf{Cgg}(\mathbf{F}^{(\ell-1)}, \mathbf{F}^{'(\ell)}), \tag{7}$$

where $\mathbf{F}^{'(\ell)}$ denotes the decoded node features at the same level of $\mathbf{F}^{(\ell)}$. As is evident, the favorable and hierarchical local context encoded in $\mathbf{F}^{'(\ell)}$ is propagated back to all nodes $\mathbf{F}^{(\ell-1)}$ at level $\ell-1$, to achieve local context decoding with compact message passing.

Notably, refer to other learning-based methods [Sun *et al.*, 2020; Zhao *et al.*, 2021], we insert inlier predictors into each HAGR module during the training phase but only preserve the last one at inference.

2.3 Orthogonal Fusion

Given a set of intermediate feature vectors $\mathcal{F} = \{\mathbf{f}_i\}_{i=1}^N$ which can be deemed as local features, the global representation vector $\mathbf{f}_g \in \mathcal{R}^C$ can be simply generated with global average pooling, which however is ill-suited for the correspondence learning task, since it treats each correspondence equally - in other words, it overlooks the negative impact of considerable outliers included in the tentative set, making the global representation vector $\mathbf{f}_g$ not robust. To tackle this issue, we propose to weigh the importance of each correspondence in light of the probability set ${}^{(t)}\mathbf{w}$ output by the t-th HAGR module, to guide the network to embed more robust global context. To this end, we extend global average pooling to a weighted formulation as follows:

$$\mathbf{f}_g = \mathcal{G}(\mathcal{F}, {}^{(t)}\mathbf{w}) = \sum_{i=1}^{N} \frac{{}^{(t)}w_i}{\sum_{j=1}^{N} {}^{(t)}w_j} \mathbf{f}_i, \tag{8}$$

where $\mathcal{G}(\cdot,\cdot)$ is the weighted global average pooling. After obtaining $\mathbf{f}_g$, we feed it into a bottleneck with two MLPs whose output channels are 32 and 128, respectively, to generate a global context-enhanced vector $\mathbf{f}_{g'}$. Then, we calculates the projection $\mathbf{f}_{i,proj}$ of each feature vector $\mathbf{f}_i$ onto the global representation $\mathbf{f}_{g'}$. Mathematically, the projection can be written as:

$$\mathbf{f}_{i,proj} = \frac{\mathbf{f}_i \cdot \mathbf{f}_{g'}}{\|\mathbf{f}_{g'}\|^2} \mathbf{f}_{g'}. \tag{9}$$

As demonstrated in Fig. 3(b), the orthogonal component is the difference between each feature vector $\mathbf{f}_i$ and its projection vector $\mathbf{f}_{i,proj}$, therefore, we can obtain the component orthogonal to $\mathbf{f}_{g'}$ by:

$$\mathbf{f}_{i,orth} = \mathbf{f}_i - \mathbf{f}_{i,proj}. \tag{10}$$

Afterwards, we append to each $\mathbf{f}_{i,orth}$ with $\mathbf{f}_{g'}$ and then the new vector is fed into an MLP whose output channel is 128. By doing so, a compact feature vector, where local and global information is well integrated, is generated. The workflow of our *orthogonal fusion* module is presented in Fig. 3(a).

2.4 Loss Formulation

Following previous studies [Yi *et al.*, 2018; Zhang *et al.*, 2019], the optimization objective of our network is to minimize a hybrid loss function as follows:

$$\mathcal{L} = \sum_{t=1}^{T} \mathcal{L}_{cls}({}^{(t)}\mathbf{w}, \mathbf{L}) + \alpha \mathcal{L}_{reg}({}^{(t)}\widehat{\mathbf{E}}, \mathbf{E}), \tag{11}$$

where superscript (t) means the t-th HAGR module, and α balances two loss terms. $\mathcal{L}_{cls}(\cdot,\cdot)$ denotes a binary cross entropy loss for the classification term:

$$\mathcal{L}_{cls}(\mathbf{w}, \mathbf{L}) = \frac{1}{N} \sum_{i=1}^{N} \mu_i \mathcal{B}(w_i, l_i), \tag{12}$$

where N is the number of tentative matches, μ_i is the per-label weight to balance positive and negative examples, $\mathcal{B}$ represents the binary cross entropy, $\mathbf{w}$ is the reasoned probability set, and $\mathbf{L} = \{l_i\}_{i=1}^N$ denotes weakly supervised labels which are generated based on the geometric error [Hartley and Zisserman, 2003] with a threshold of 10^{-4}. $\mathcal{L}_{reg}(\cdot,\cdot)$ is the regression loss between the estimated essential matrix $\widehat{\mathbf{E}}$ and the ground-truth one $\mathbf{E}$, which is also based on the geometric error:

$$\begin{aligned}&\mathcal{L}_{reg}(\widehat{\mathbf{E}}, \mathbf{E}) \\ &= \sum_{i=1}^{N} \frac{(\mathbf{p}_i'^{\top} \widehat{\mathbf{E}} \mathbf{p}_i)^2}{\|\mathbf{E}\mathbf{p}_i\|_{[1]}^2 + \|\mathbf{E}\mathbf{p}_i\|_{[2]}^2 + \|\mathbf{E}\mathbf{p}_i'\|_{[1]}^2 + \|\mathbf{E}\mathbf{p}_i'\|_{[2]}^2},\end{aligned} \tag{13}$$

where $\mathbf{p}_i$ and $\mathbf{p}_i'$ are two keypoints forming the correspondence $\mathbf{c}_i$, and $\mathbf{v}_{[i]}$ denotes the i-th element of vector $\mathbf{v}$.

2.5 Implementation Details

In our implementation, the input to our model is $N \times 4$ putative correspondences established by an NN matcher with SIFT detector-descriptors, typically $N = 2000$, unless otherwise specified. The number of levels is set to $L = 4$, *i.e.,* each HRGA module contains three LCPool layers with the sampling ratios of 0.125, 0.5, 0.5, respectively. We use 4-head attention in the context aggregation layer. We implement our model with Pytorch and adopt Adam optimizer with a learning rate of 10^{-4} and a batch size of 32 in optimization. Weight α is set to 0 at the start and to 0.5 after first 20k iterations. All experiments are conducted on Ubuntu 18.04 with GeForce RTX 3090 GPUs.

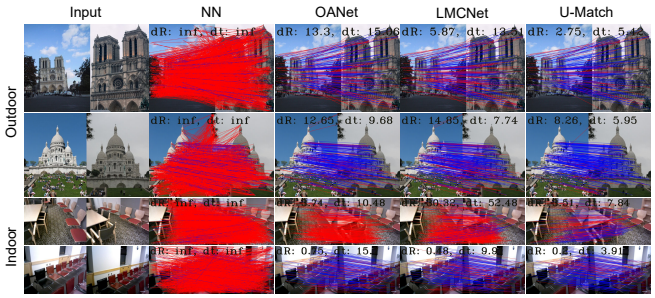

Figure 4: Matching results on YFCC100M and SUN3D with SIFT features. Correspondences are in blue if they are consistent with the ground-truth epipolar geometry, and in red otherwise. Pose estimation results are shown at the top left corner.

3 Experiments

In the following sessions, we first evaluate U-Match on three diverse problems which heavily rely on two-view correspondence learning, namely: relative pose estimation, homography estimation, and visual localization. Then, a comprehensive analysis is provided for better understanding U-Match.

3.1 Relative Pose Estimation

Relative pose estimation, which refers to accurately estimate the relative position relationship (*i.e.*, rotation and translation) between different camera views with identified inliers, plays a pivotal role in computer vision.

Datasets. As in the previous work [Zhang *et al.*, 2019], we resort to two popular datasets, YFCC100M [Thomee *et al.*, 2016] and SUN3D [Xiao *et al.*, 2013], to demonstrate the correspondence learning ability of our method in outdoor and indoor scenes, respectively. YFCC100M contains 100 million images from Internet, which are split into 72 sequences according to different tourist spots. We choose 68 sequences as training and validation data, and the remaining sequences are used for testing. SUN3D is a large-scale RGB-D video dataset with relative camera motions retrieved by generalized bundle adjustment. It is comprised of 254 indoor image sequences with poor texture, repetitive elements, and self-occlusions, where 239 sequences are adopted for training and validation, and the rest sequences are used for testing.

Evaluation Protocols. We report the area under the cumulative error curve (AUC) of the pose error at multiple thresholds (5°, 10°, 20°), where the pose error is the maximum of the angular error in rotation and translation. Importantly, relative poses are recovered by estimating essential matrix with both the weighted eight-point algorithm and RANSAC.

Baselines. We compare U-Match with 1) traditional handcrafted approaches, including the NN matcher, USAC [Raguram *et al.*, 2012], VFC [Ma *et al.*, 2014], LPM [Ma *et al.*, 2019], and GMS [Bian *et al.*, 2017], 2) learning-based ones, including PointCN [Yi *et al.*, 2018], OANet [Zhang *et al.*, 2019], CLNet [Zhao *et al.*, 2021], LMCNet [Liu *et al.*, 2021], and MS^{2}DG-Net [Dai *et al.*, 2022]. To provide a fair evaluation, all learning-based methods are re-trained in the same training setting. Considering that SuperGlue [Sarlin *et al.*, 2020] is the most popular learning-based feature matching method, we also deliver the results of SIFT+SuperGlue on YFCC100M with RANSAC.

Method	YFCC100M (outdoor) (%)		
	AUC@5°	AUC@10°	AUC@20°
PointCN [Yi *et al.*, 2018]	25.72	37.73	46.23
OANet [Zhang *et al.*, 2019]	39.05	53.69	62.12
CLNet [Zhao *et al.*, 2021]	51.92	63.11	69.85
LMCNet [Liu *et al.*, 2021]	50.30	62.56	69.62
MS^{2}DG-Net [Dai *et al.*, 2022]	48.38	62.45	70.52
U-Match	**60.53**	**71.26**	**80.37**

Table 1: Evaluation on YFCC100M for outdoor pose estimation with the weighted eight-point algorithm.

Method	YFCC100M (outdoor) (%)		
	AUC@5°	AUC@10°	AUC@20°
NN	9.18	14.64	19.28
USAC [Raguram *et al.*, 2012]	5.67	9.53	13.41
VFC [Ma *et al.*, 2014]	33.70	41.29	46.27
LPM [Ma *et al.*, 2019]	30.78	39.31	44.81
GMS [Bian *et al.*, 2017]	26.30	34.59	40.43
PointCN [Yi *et al.*, 2018]	50.32	59.88	65.83
OANet [Zhang *et al.*, 2019]	52.42	62.79	68.86
CLNet [Zhao *et al.*, 2021]	59.05	69.03	74.97
LMCNet [Liu *et al.*, 2021]	57.57	67.40	73.07
MS^{2}DG-Net [Dai *et al.*, 2022]	58.05	68.34	74.33
U-Match	**60.38**	**70.05**	**78.86**
SuperGlue* [Sarlin *et al.*, 2020]	59.25	77.41	85.70

Table 2: Evaluation on YFCC100M for outdoor pose estimation with RANSAC. The results of SuperGlue are cited from its supplementary material.

Results of Outdoor Pose Estimation. As presented in the first two rows of Fig. 4, several visualization results of outdoor scenes reveal that the proposed U-Match is capable of rejecting spurious correspondences while preserving correct ones effectively. The quantitative results with the weighted eight-point algorithm and RANSAC are reported in Tables 1 and 2, respectively. Clearly, U-Match outperforms other baselines at all error thresholds by a significant margin in both evaluation modes, even surpassing SuperGlue, which requires extra visual descriptors as input, in terms of AUC@5°. Surprisingly, U-Match is the first model to pass the 60% AUC@5° bar, which without RANSAC even excels all prior models with RANSAC at each threshold. We attribute the top performance to implicit multi-level local context captured in an encoder-decoder manner with the HAGR module. Moreover, the OF module also contributes to the estimation accuracy by integrating complementary local and global context without redundancy.

Results of Indoor Pose Estimation. Compared to outdoor scenes, texture-less scenes of indoor environments are more challenging to relative pose estimation. Even so, visualization shown in the last two rows of Fig. 4 qualitatively demonstrates that U-Match can achieve impressive performance. Further looking at Tables 3 and 4, as with YFCC100M, U-Match consistently achieves the best accuracy at all AUC thresholds, irrespective of using the weighted eight-point algorithm or RANSAC. Intriguingly, U-Match

Method	SUN3D (indoor) (%)		
	AUC@5°	AUC@10°	AUC@20°
PointCN	8.81	16.88	24.19
OANet	15.60	25.83	40.07
CLNet	9.58	17.98	25.42
LMCNet	18.95	30.33	38.94
MS^{2}DG-Net	16.10	27.01	35.57
U-Match	**21.46**	**32.68**	**47.13**

Table 3: Evaluation on SUN3D for indoor pose estimation with the weighted eight-point algorithm.

Method	SUN3D (indoor) (%)		
	AUC@5°	AUC@10°	AUC@20°
NN	3.07	5.81	8.57
USAC [Raguram *et al.*, 2012]	3.30	5.99	8.73
VFC [Ma *et al.*, 2014]	13.39	20.42	25.96
LPM [Ma *et al.*, 2019]	12.51	19.35	24.79
GMS [Bian *et al.*, 2017]	10.58	16.63	21.59
PointCN [Yi *et al.*, 2018]	15.50	24.12	31.01
OANet [Zhang *et al.*, 2019]	16.76	26.07	38.40
CLNet [Zhao *et al.*, 2021]	16.30	25.89	33.15
LMCNet [Liu *et al.*, 2021]	17.83	27.61	35.06
MS^{2}DG-Net [Dai *et al.*, 2022]	18.18	27.85	35.21
U-Match	**18.34**	**28.07**	**41.22**

Table 4: Evaluation on SUN3D for indoor pose estimation with RANSAC.

without RANSAC is still more powerful than other baselines with RANSAC according to all metrics.

3.2 Homography Estimation

Homography estimation aims to find a linear image-to-image mapping in the homogeneous space, acting as a crucial prerequisite for a board range of downstream applications. In the following, we evaluate the performance of U-Match with both robust (RANSAC) and non-robust Direct Linear Transform (DLT) estimators on this task.

Dataset. We resort to the HPatches dataset [Balntas *et al.*, 2017] for evaluation, which contains 52 sequences changing largely in viewpoint and 56 sequences varying significantly in illumination conditions. Each consists of one reference image and five query ones, with ground-truth homographies. Importantly, we extract up to 4000 keypoints with SIFT followed by an NN matcher to establish tentative matches.

Evaluation Protocols. Following the corner correctness metric used in [DeTone *et al.*, 2018], we report the percentage of correctly estimated homographies whose average error is below 3/5/10 pixels.

Results. Table 5 presents the all-sided quantitative evaluation on HPatches. Clearly, U-Match achieves best results at all thresholds, no matter with DLT or RANSAC estimators, proving that our method is well-suited for this task.

3.3 Visual Localization

Given a query image, visual localization aims to estimate its 6-DOF camera pose with respect to the corresponding 3D scene model. In the following, we integrate U-Match into the official HLoc [Sarlin *et al.*, 2019] pipeline to investigate how our model directly benefits the visual localization task.

Method	HPathces (%)		
	ACC.@3px	ACC.@5px	ACC.@10px
PointCN [Yi *et al.*, 2018]	38.97/67.93	51.55/82.59	65.34/92.76
OANet [Zhang *et al.*, 2019]	39.83/69.66	52.76/82.93	62.93/91.90
CLNet [Zhao *et al.*, 2021]	43.10/57.07	55.69/73.45	68.10/86.90
LMCNet [Liu *et al.*, 2021]	47.76/**72.93**	58.79/83.62	70.00/92.76
MS^{2}DG-Net [Dai *et al.*, 2022]	41.21/72.07	50.17/83.28	62.59/92.62
U-Match	**48.90/72.93**	**59.41/84.48**	**70.83/92.90**

Table 5: Evaluation on HPatches for homography estimation. The percentage of correctly estimated homographies, *i.e.*, accuracy (*ACC.*), at different error thresholds (**without/with** RANSAC post-processing) is reported.

Dataset. We adopt Aachen Day-Night benchmark [Sattler *et al.*, 2018] which contains 4328 reference images from a European old town and 922 (824 daytime, 98 nighttime) query images taken by mobile phone cameras, to validate the effectiveness of our network on visual localization.

Evaluation Protocols. Consistent with the official benchmark, we report the percentage of correctly localized queries at specific distance and orientation thresholds. Importantly, we extract up to 4096 keypoints per image with SIFT, establish putative correspondences with an NN matcher, triangulate an SfM model from day-time images with known poses, and register night-time query images with 2D-2D matches provided by correspondence learning methods and COLMAP [Schonberger and Frahm, 2016].

Results. As summarized in Table 7, the results show that our approach notably performs on par with or better than its competitors in both day and night scenes at all error thresholds, demonstrating its superiority on this task.

3.4 Understanding U-Match

Impact of Graph Representation Levels. Intuitively, the HAGR module with larger levels enables more complex local context awareness. To investigate the impact of graph representation levels, we further train our U-Match on YFCC100M with 2, 3, 5 levels, respectively. As rows 4-7 shown in Table 8, the models with 2 or 3 levels lead to substantially worse results, since underlying hierarchical local context cannot be fully explored with fewer levels. Also, the 5-level model performs on par with the default model (*i.e.*, 4 levels), indicating that too many layers are unnecessary since the unearthed local context might be redundant.

Efficiency. For an HAGR module, let $k^{(1)}$, $k^{(2)}$, and $k^{(3)}$ denote the number of sampled nodes at three levels, respectively, the theoretical complexity of the pooling and unpooling operations at three levels and the context aggregation layer at the bottom is $O(Nk^{(1)})$, $O(k^{(1)}k^{(2)})$, $O(k^{(2)}k^{(3)})$, and $O({k^{(3)}}^2)$, respectively. Since $k^{(1)}, k^{(2)}, k^{(3)} \ll N$, the complexity of an HAGR module can be approximately written as $O(N)$. Table 6 summarizes the average runtime of all learning-based methods for correspondence learning on YFCC100M using a single GeForce RTX 3090 GPU with

Method	PointCN [Yi *et al.*, 2018]	OANet [Zhang *et al.*, 2019]	CLNet [Zhao *et al.*, 2021]	LMCNet [Liu *et al.*, 2021]	MS^{2}DG-Net [Dai *et al.*, 2022]	**U-Match**
ART	**19.85**/124.07	35.25/170.79	39.55/**63.72**	247.53/385.15	37.75/167.60	55.50/152.00

Table 6: Efficiency evaluation. Average runtime (ART, unit: ms) on YFCC100M (**without/with** RANSAC post-processing) is reported.

Method	Day	Night
	(0.25m, 2°) / (0.5m, 5°) / (1.0m, 10°)	
PointCN [Yi *et al.*, 2018]	81.3/91.4/95.9	68.4/78.6/87.8
OANet [Zhang *et al.*, 2019]	82.3/91.9/96.5	71.4/79.6/90.8
CLNet [Zhao *et al.*, 2021]	83.3/92.4/**97.0**	71.4/80.6/**93.9**
MS^{2}DG-Net [Dai *et al.*, 2022]	82.8/92.1/96.8	70.4/**82.7**/**93.9**
U-Match	**85.3**/**92.6**/96.8	**72.4**/**82.7**/90.8

Table 7: Evaluation on Aachen Day-Night for visual localization. The percentage of correctly localized queries at different thresholds is reported.

Method	YFCC100M (outdoor) (%)		
	AUC@5°	AUC@10°	AUC@20°
(1) U-Match w. Concatenation	58.33	69.33	78.77
(2) U-Match w. Hadamard	59.60	70.18	79.47
(3) U-Match w/o. OF	57.53	68.34	77.75
(4) U-Match w. 2 levels	54.95	66.63	77.19
(5) U-Match w. 3 levels	56.93	68.18	78.28
(6) U-Match w. 5 levels	**60.70**	70.94	80.23
(7) U-Match full	60.53	**71.26**	**80.37**

Table 8: Ablation study of U-Match. w. Concatenation and w. Hadamard stand for replacing the orthogonal decomposition step with concatenation and Hadamard product, respectively. w/o. OF stands for removing the *orthogonal fusion* module from U-Match. AUC@5° (%) with the weighted eight-point algorithm is reported.

24GB memory. Clearly, U-Match achieves comparable efficiency regardless of using the weighted eight-point algorithm or RANSAC. In contrast, LMCNet is time-consuming, whose average runtime without RANSAC is an order of magnitude larger than others due to the calculation of smooth motions via the coherence residual layer. CLNet with RANSAC runs the fastest since it just employs identified inliers from the pruned candidates to estimate the essential matrix.

Generalization Ability. To investigate the generalization ability of U-Match to different combinations of datasets and detector-descriptors, we test all learning-based counterparts on YFCC100M with RootSIFT or SuperPoint, as well as SUN3D with SIFT, RootSIFT or SuperPoint, adopting the models only trained on YFCC100M with SIFT. Notably, we extract up to 2000 and 1000 keypoints per image with RootSIFT and SuperPoint, respectively, followed by an NN matcher to generate putative matches. As reported in Table 9, U-Match substantially outperforms prior state-of-the-arts in all cases, directly reflecting its superior generalization ability.

Ablation Study. In the OF module, we propose to decompose a local feature into two parts, namely $\mathbf{f}_{i,proj}$ and $\mathbf{f}_{i,orth}$, where the former is parallel to the global feature $\mathbf{f}_{g'}$ and the latter is orthogonal to $\mathbf{f}_{g'}$. To demonstrate that such operation is a better choice, we replace the orthogonal decomposition step with concatenation and Hadamard product, respectively,

Method	YFCC100M (outdoor) (%)		SUN3D (indoor) (%)		
	RootSIFT	SuperPoint	SIFT	RootSIFT	SuperPoint
PointCN	25.60/50.98	14.70/41.10	1.14/14.94	1.26/15.26	3.15/14.05
OANet	40.25/55.75	20.95/43.38	2.80/14.03	2.92/14.58	2.87/13.84
CLNet	51.35/60.77	24.68/42.70	2.28/15.49	2.43/16.24	2.86/9.61
LMCNet	50.48/58.58	24.88/42.83	4.79/14.97	5.04/15.38	3.22/14.23
MS^{2}DG-Net	50.05/59.62	25.75/44.60	4.72/15.64	5.07/15.86	3.40/14.64
U-Match	**61.32/60.88**	**28.38/45.12**	**6.66/15.93**	**6.59/16.26**	**3.87/14.84**

Table 9: Generalization ability test. AUC@5° (%) (**without/with** RANSAC post-processing) is reported.

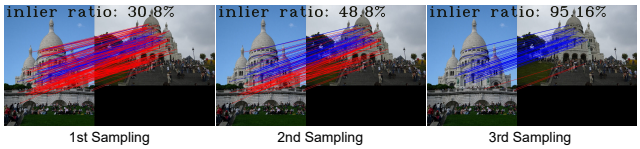

Figure 5: Visualization of sampled nodes by three LCPool layers.

which are commonly used to fuse two feature vectors. As rows 1, 2 and 7 reported in Table 8, the proposed orthogonal fusion achieves the best performance among three fusion strategies, since by doing so, there is no information redundancy in the output local feature $\mathbf{f}_{i,orth}$. In addition, row 3 verifies the effectiveness of the propose OF module to combine complementary local and global context.

Visualization of Sampled Nodes. We visualize the sampled correspondences at three levels, as shown in Fig. 5. U-Match is capable of progressively discovering reliable matches as message bottlenecks, to achieve accurate and robust local context encoding, as evidenced by the gradually increasing inlier ratio of sampled nodes.

4 Conclusion

This paper introduces U-Match for two-view correspondence learning. The main improvement is from two sides: 1) We propose an efficient network structure in an encoder-decoder manner, implicitly capturing the local context from different levels and can be trained integrally. 2) We design an orthogonal fusion module that combines complementary local and global context without redundancy. Experiments on different tasks and datasets show that U-Match brings significant improvement over the state-of-the-art methods.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (62276192), and the Key Research and Development Program of Hubei Province (2020BAB113).

Contribution Statement

Zizhuo Li and Shihua Zhang contributed equally to this work.

References

[Balntas *et al.*, 2017] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proc. CVPR*, pages 5173–5182, 2017.

[Barath *et al.*, 2020] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *Proc. CVPR*, pages 1304–1312, 2020.

[Bian *et al.*, 2017] JiaWang Bian, Wen-Yan Lin, Yasuyuki Matsushita, Sai-Kit Yeung, Tan-Dat Nguyen, and Ming-Ming Cheng. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *Proc. CVPR*, pages 4181–4190, 2017.

[Dai *et al.*, 2022] Luanyuan Dai, Yizhang Liu, Jiayi Ma, Lifang Wei, Taotao Lai, Changcai Yang, and Riqing Chen. Ms2dg-net: Progressive correspondence learning via multiple sparse semantics dynamic graph. In *Proc. CVPR*, pages 8973–8982, 2022.

[DeTone *et al.*, 2018] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proc. CVPR*, pages 224–236, 2018.

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

[Hartley and Zisserman, 2003] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[Liu *et al.*, 2021] Yuan Liu, Lingjie Liu, Cheng Lin, Zhen Dong, and Wenping Wang. Learnable motion coherence for correspondence pruning. In *Proc. CVPR*, pages 3237–3246, 2021.

[Lowe, 2004] David G Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004.

[Ma *et al.*, 2014] Jiayi Ma, Ji Zhao, Jinwen Tian, Alan L Yuille, and Zhuowen Tu. Robust point matching via vector field consensus. *IEEE Trans. Image Process.*, 23(4):1706–1721, 2014.

[Ma *et al.*, 2019] Jiayi Ma, Ji Zhao, Junjun Jiang, Huabing Zhou, and Xiaojie Guo. Locality preserving matching. *Int. J. Comput. Vis.*, 127(5):512–531, 2019.

[Ma *et al.*, 2021] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image matching from handcrafted to deep features: A survey. *Int. J. Comput. Vis.*, 129(1):23–79, 2021.

[Qi *et al.*, 2017] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. CVPR*, pages 652–660, 2017.

[Raguram *et al.*, 2012] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. Usac: A universal framework for random sample consensus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):2022–2038, 2012.

[Sarlin *et al.*, 2019] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proc. CVPR*, pages 12716–12725, 2019.

[Sarlin *et al.*, 2020] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proc. CVPR*, pages 4938–4947, 2020.

[Sattler *et al.*, 2018] 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 *Proc. CVPR*, pages 8601–8610, 2018.

[Schonberger and Frahm, 2016] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. CVPR*, pages 4104–4113, 2016.

[Sun *et al.*, 2020] Weiwei Sun, Wei Jiang, Eduard Trulls, Andrea Tagliasacchi, and Kwang Moo Yi. Acne: Attentive context normalization for robust permutation-equivariant learning. In *Proc. CVPR*, pages 11286–11295, 2020.

[Tang *et al.*, 2022] Linfeng Tang, Yuxin Deng, Yong Ma, Jun Huang, and Jiayi Ma. Superfusion: A versatile image registration and fusion network with semantic awareness. *IEEE/CAA J. Autom. Sinica*, 9(12):2121–2137, 2022.

[Thomee *et al.*, 2016] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Commun. ACM*, 59(2):64–73, 2016.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30:1–11, 2017.

[Xiao *et al.*, 2013] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proc. ICCV*, pages 1625–1632, 2013.

[Yi *et al.*, 2018] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proc. CVPR*, pages 2666–2674, 2018.

[Zhang *et al.*, 2019] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *Proc. ICCV*, pages 5845–5854, 2019.

[Zhao *et al.*, 2021] Chen Zhao, Yixiao Ge, Feng Zhu, Rui Zhao, Hongsheng Li, and Mathieu Salzmann. Progressive correspondence pruning by consensus learning. In *Proc. ICCV*, pages 6464–6473, 2021.