Boosting Correspondence Learning with Structure-Aware Estimator

Tianyu Yan, Wei An, Pu Wang*, and Yingqian Wang*

National University of Defense Technology
yantianyu@nudt.edu.cn

Abstract. Correspondence learning aims to estimate accurate geometric model parameters from the data with severe outliers and is crucial for many computer vision tasks. While recent deep correspondence learning methods have substantially improved inlier-outlier identification, most of them still rely on a differentiable weighted least squares (WLS) estimator to recover the final model parameters. This strategy typically treats correspondences as conditionally independent observations and therefore ignores the structured dependencies among inliers. In real-world scenarios, inliers are often sampled from geometric manifolds and tend to appear in spatially dense clusters, especially in texture-rich regions. As a result, WLS may be dominated by dense inlier groups, leading to a poorly-conditioned estimation problem that is vulnerable to noise or degenerate configurations. To address this bottleneck, we propose a structure-aware estimator, which explicitly models inlier correlations via a graph Laplacian matrix and integrates this prior into a maximum-likelihood framework. Our proposed estimator is fully differentiable, architecturally lightweight, and can seamlessly replace the standard WLS estimator in existing correspondence learning pipelines. Extensive experiments on multiple benchmarks demonstrate consistent improvements while introducing a modest increase in computational cost. The code is publicly available at: https://github.com/Tianyu-Yan/SAE

Keywords: Correspondence learning · Graph structure · WLS

1 Introduction

Estimating accurate geometric model parameters from correspondences with severe outliers constitutes a foundational task in computer vision, serving as a basic step for relative pose estimation, point cloud registration, simultaneous localization and mapping, and visual localization tasks [12]. The standard pipeline typically begins with detecting keypoints and establishing putative correspondences via descriptor matching [9,15,30]. However, in real-world scenarios characterized by extreme viewpoint changes, repetitive textures, partial occlusions or illumination variations, this set of putative correspondences is inevitably contaminated by a significant proportion of outliers [17]. Consequently, the correspondence

* corresponding authors

learning task aims to robustly estimate the geometric model parameters from these noisy and outlier-prone data.

Recent progress has been driven by deep learning networks [14, 25, 31, 32, 34, 37] that substantially surpass traditional methods [3–5, 10, 20]. By treating the task as an inlier classification and geometric model estimation problem, these learning-based methods have demonstrated remarkable success in extracting rich geometric contexts and accurately classifying inliers by employing diverse neural architectures, including permutation-equivariant networks [28, 31, 32], convolutional backbones [16, 33, 35], graph-based operations [8, 37], and attention (*e.g.*, transformer) mechanisms [14, 18].

Despite the strong feature representations learned by these networks, the geometric model estimation stage in correspondence learning pipelines almost relies on a differentiable weighted least squares (WLS) estimator [27] (*e.g.*, weighted eight-point algorithm [21]). However, the WLS inherently imposes a strict assumption of *conditional independence* among all observations, which is not fully aligned with the practical scenes. In real-world scenarios, inliers are not independently scattered. They are typically sampled from continuous physical manifolds (*e.g.*, planar surfaces or rigid objects) and tend to cluster in texture-rich regions or coherent scene structures [17] (*i.e.*, inliers exhibit probabilistic dependencies). This causes densely grouped inliers to disproportionately dominate the optimization in WLS, leading to a poorly-conditioned estimation problem that is vulnerable to local noise and easily trapped in degenerate configurations [12]. Therefore, better feature learning designs alone cannot overcome the bottleneck in WLS due to its independence assumption.

To bridge this gap, we introduce a novel structure-aware estimator grounded in the theory of Gaussian Markov random fields (GMRF) [22]. Instead of assuming independence, we leverage the graph Laplacian matrix to explicitly model the probabilistic correlations among inliers. Integrating this prior into a maximum likelihood framework yields a differentiable and closed-form estimator. Crucially, our proposed estimator serves as a lightweight module that can replace the standard WLS estimator in most existing correspondence learning networks. This plug-in design consistently improves model estimation performance while introducing only a modest computational overhead.

Our main contributions are summarized as follows:

- We propose a fully differentiable, computationally efficient structure-aware estimator, that can be seamlessly integrated into most correspondence learning networks and boost their performance.
- We explicitly model correlations among inliers via GMRF, breaking the conditional independence assumption inherent in the WLS estimator.
- Extensive experiments on multiple benchmarks consistently demonstrate that existing correspondence learning networks, when equipped with our estimator, achieve significant performance gains with only a modest increase in computational cost.

2 Related Work

2.1 Traditional Methods

The dominant paradigm has long been established upon random sample consensus (RANSAC) [10] and its extensions. Operating on a hypothesize-and-verify principle, RANSAC iteratively samples minimal data subsets to generate parametric models and identifies the consensus set that maximizes model support. Over the decades, numerous variants have been proposed to enhance its efficiency and robustness. For instance, LO-RANSAC [5] introduces local optimization steps to refine the hypotheses, while USAC [20] integrates multiple advanced sampling and verification strategies into a unified framework. More recently, MAGSAC [3] and MAGSAC++ [4] have mitigated the sensitivity to user-defined inlier thresholds by marginalizing over continuous noise scales. Despite these advancements, the fundamental limitation of RANSAC families remains their computational complexity, which scales exponentially with the outlier ratio. Furthermore, their non-differentiable nature hinders seamless end-to-end integration with deep-learning based pipelines.

2.2 Learning-Based Methods

To address the limitations of handcrafted heuristics, recent research has shifted toward data-driven correspondence learning. PointCN [31] pioneered this direction by applying permutation-equivariant multi-layer perceptrons (MLPs) to process unordered correspondences. To better capture geometric contexts, subsequent architectures such as OANet [32] and T-Net++ [28] incorporate sophisticated permutation-equivariant structures with local-global attention fusion to fully exploit contextual information. Beyond MLP-based networks, ConvMatch [35] and DeMo [16] formulate the task within a regularized motion field, utilizing CNNs to extract spatial motion features. More recent methods like DeMatch [33] and DeMatch++ [34] decompose motion fields into smooth sub-fields for implicit regularization. Despite the remarkable success of the aforementioned networks in correspondence learning, they universally rely on a simple WLS estimator (*e.g.*, the weighted eight-point algorithm [21]). The WLS fundamentally ignores the complex probabilistic correlations among inliers, thereby bottlenecking the theoretical upper limit of the estimation accuracy.

2.3 Graph-Related Correspondence Learning

Many methods adopt graph neural networks (GNNs) to explicitly formulate correspondence structures, significantly improving inlier classification. Early methods like MS^{2}DG-Net [8] and CLNet [37] construct nearest-neighbor graphs in the coordinate or feature space, employing graph convolutions to aggregate local and global consistency. Recent advances have focused on refining graph construction and message-passing mechanisms. For example, BCLNet [18] establishes a

global-graph space to explicitly capture the relationships between correspondences. Other methods, like NCMNet [14], MGNet [7], and TRGANet [6], leverage multi-graph consensus and cross-stage contextual attention to maintain geometric consistency. Although these graph-based methods successfully leverage topology to learn powerful representations for inlier classification, they still use the standard WLS estimator for the final model estimation, without explicitly accounting for the probabilistic dependencies among inliers during the optimization phase. In this paper, we directly map the topological information into the estimator itself, achieving a mathematically principled and fully differentiable structure-aware estimator.

3 Background

Given a set of observations $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_N\}$, we aim to recover the underlying geometric model, such as the fundamental matrix $\mathbf{F} \in \mathbb{R}^{3\times3}$ or homography $\mathbf{H} \in \mathbb{R}^{3\times3}$. In the outlier-free case, due to the simplicity and efficiency of algebraic error, most of such estimation problems can be formulated in the following form and easily solved via direct linear transformation (DLT) [12]:

$$\mathbf{z}^* = \arg\min_{\|\mathbf{z}\|=1} \|\mathbf{M}\mathbf{z}\|^2, \tag{1}$$

where $\mathbf{M}$ is the data matrix, $\mathbf{z}$ is the underlying geometric model parameters, and $\mathbf{z}^*$ is the optimal estimate. The constraint $\|\mathbf{z}\| = 1$ is to avoid trivial solutions such as $\mathbf{z} = \mathbf{0}$. In the *supplementary material*, we provide details how to construct $\mathbf{M}$ and derive the form of Eq. (1) for many tasks such as fundamental matrix estimation, homography estimation, and line fitting. According to the above optimization problem, the estimated geometric model parameters are given as the right singular vector corresponding to the smallest singular value of $\mathbf{M}$. In the absence of outliers, the DLT typically approximates the true solution closely. However, real-world data invariably contains outliers. Consequently, relying on the DLT can lead to significantly erroneous estimation, as the method is highly sensitive to even a single outlier.

When considering outliers, a practical way to estimate geometric model parameters is by solving a sequence of weighted least squares [21, 27] problems:

$$\mathbf{z}_j = \arg\min_{\|\mathbf{z}\|=1} \mathbf{z}^T\mathbf{M}^T\mathbf{W}\mathbf{M}\mathbf{z}, \tag{2}$$

where $\mathbf{z}_j$ represents the estimated parameters and $\mathbf{W} = \mathrm{diag}(w_1, \cdots, w_N)$ denotes the diagonal weight matrix at the j-th iteration. Ideally, the algorithm assigns very low weights (close to zero) to outliers to suppress their influence on model estimation, while assigning relatively larger weights to inliers. This paradigm is widely used in many geometric model estimation problems, such as iteratively reweighted least squares (IRLS) for M-estimator [11] and weighted eight-points method for learning based fundamental matrix estimation [21].

The above optimization can also be explained from a Bayesian perspective. If we assume the residuals of observations $\mathbf{r} = \mathbf{Mz}$ follows a Gaussian distribution with zero mean, we can obtain the following likelihood function:

$$p(\mathbf{r}|\mathbf{z}) = \frac{1}{(2\pi)^{N/2}\sqrt{\mathbf{\Omega}}} \exp\left(-\frac{1}{2}\mathbf{z}^T\mathbf{M}^T\mathbf{\Omega}^{-1}\mathbf{Mz}\right), \tag{3}$$

where $\mathbf{r} \sim \mathcal{N}(0, \mathbf{\Omega})$ and $\mathbf{\Omega} = \text{diag}\{\sigma_1^2, \sigma_2^2, \cdots, \sigma_N^2\}$ is the covariance matrix with heteroscedastic Gaussian noise, the maximum likelihood estimation (MLE) can directly lead to Eq. (2), where $\mathbf{W} = \mathbf{\Omega}^{-1} = \text{diag}(1/\sigma_1^2, \cdots, 1/\sigma_N^2)$.

4 Method

As discussed in the background, the existing WLS estimator associates each observation with an individual reliability measure, effectively suppressing outliers by adjusting variances. From a Bayesian perspective, it imposes a strict constraint of *conditional independence* among all observations.

However, in real-world scenarios, inliers are always sampled from continuous physical manifolds and tend to cluster in textured regions. Consequently, local physical perturbations, such as lens distortion, systematic sensor noise, or local illumination variations, cause neighboring inliers to exhibit highly correlated errors, which implies that inliers exhibit probabilistic dependencies.

To capture these probabilistic dependencies, we argue that the inlier set should not be treated as a collection of isolated measurements, but rather as an undirected graph structure, where edges encode the probabilistic dependencies. To seamlessly incorporate this graph topology into the parameter estimation process, we propose to model these inlier residual vectors via GMRF [22]. In the following sections, we first derive the closed-form estimator and then demonstrate its integration into correspondence learning pipelines.

4.1 Closed-Form Estimator

A straightforward application of a standard GMRF to the residual vector $\mathbf{r} \in \mathbb{R}^N$ would assume homoscedasticity (*i.e.*, uniform variance across all observations), which is inconsistent with the reality that the measurement uncertainties σ_i of observations are different. To address this, we mathematically decouple the individual measurement uncertainty from the geometric structural correlation. In detail, we formulate the observed residual vector $\mathbf{r}$ as the product of a weight matrix and a latent structural variable:

$$\mathbf{r} = \boldsymbol{\Lambda}\boldsymbol{\eta}, \tag{4}$$

where $\boldsymbol{\Lambda} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_N)$ is a diagonal matrix capturing the individual measurement uncertainty, and $\boldsymbol{\eta} \in \mathbb{R}^N$ is the standardized variable representing the latent structural correlation.

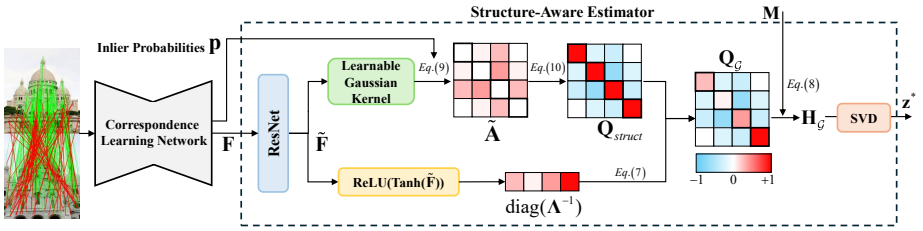

Fig. 1: Overview of the structure-aware estimator.

Instead of assuming $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ as in standard WLS, we model $\boldsymbol{\eta}$ as a GMRF over a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The motivation for adopting the GMRF framework lies in its elegant ability to map the graph Laplacian matrix directly into a sparse precision matrix, which builds a bridge between graph structure and probabilistic space. The probability density function of $\boldsymbol{\eta}$ is given by:

$$p(\boldsymbol{\eta}) \propto \exp\left(-\frac{1}{2}\boldsymbol{\eta}^T \mathbf{Q}_{struct}\boldsymbol{\eta}\right), \tag{5}$$

where $\mathbf{Q}_{struct} \in \mathbb{R}^{N\times N}$ is the precision matrix (*i.e.*, the inverse covariance matrix) defined by the graph Laplacian matrix.

According to $\boldsymbol{\eta} = \boldsymbol{\Lambda}^{-1}\mathbf{r}$, we derive the likelihood function of the observed residuals as:

$$p(\mathbf{r}|\mathbf{z}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{\Lambda}^{-1}\mathbf{r})^T \mathbf{Q}_{struct}(\boldsymbol{\Lambda}^{-1}\mathbf{r})\right). \tag{6}$$

Clearly, the residual vector follows $\mathbf{r} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\mathcal{G}}^{-1})$, where the new precision matrix $\mathbf{Q}_{\mathcal{G}}$ of $\mathbf{r}$ is defined as follows:

$$\mathbf{Q}_{\mathcal{G}} = \boldsymbol{\Lambda}^{-1}\mathbf{Q}_{struct}\boldsymbol{\Lambda}^{-1}. \tag{7}$$

Remark: It is worth noting that if the graph contains no edges (*i.e.*, no probabilistic dependencies), the precision matrix degenerates to the identity matrix ($\mathbf{Q}_{struct} = \mathbf{I}$). In this case, $\mathbf{Q}_{\mathcal{G}} = \boldsymbol{\Lambda}^{-2} = \mathrm{diag}(1/\sigma_1^2, \ldots, 1/\sigma_N^2)$, which perfectly reduces to the traditional WLS weight matrix $\mathbf{W}$. Thus, our formulation is a generalization of WLS.

Substituting $\mathbf{r} = \mathbf{M}\mathbf{z}$ into the negative log-likelihood of Eq. (6), our new optimization objective is formulated as:

$$\arg\min_{\|\mathbf{z}\|=1} \mathbf{z}^T\left(\mathbf{M}^T\boldsymbol{\Lambda}^{-1}\mathbf{Q}_{struct}\boldsymbol{\Lambda}^{-1}\mathbf{M}\right)\mathbf{z} \quad \Rightarrow \quad \arg\min_{\|\mathbf{z}\|=1} \mathbf{z}^T\mathbf{H}_G\mathbf{z}, \tag{8}$$

where $\mathbf{H}_{\mathcal{G}} = \mathbf{M}^T\mathbf{Q}_{\mathcal{G}}\mathbf{M} = \mathbf{M}^T\boldsymbol{\Lambda}^{-1}\mathbf{Q}_{struct}\boldsymbol{\Lambda}^{-1}\mathbf{M}$. The optimal geometric model parameters $\mathbf{z}^*$ is simply the eigenvector corresponding to the smallest eigenvalue of the information matrix $\mathbf{H}_{\mathcal{G}}$, which can be efficiently solved by singular value decomposition (SVD).

Algorithm 1 Structure-Aware Estimator

Require: Putative correspondences $\mathbf{S}$, network features $\mathbf{F} = \{\mathbf{f}_i\}_{i=1}^N$, predicted inlier probabilities $\mathbf{p} = \{p_i\}_{i=1}^N$

Construct data matrix $\mathbf{M}$ based on $\mathbf{S}$.
$\tilde{\mathbf{f}}_i \leftarrow \text{ResNet}(\mathbf{f}_i)$ for all $i \in \{1, \dots, N\}$
$A_{ij} \leftarrow \exp\left(-\frac{\|\tilde{\mathbf{f}}_i - \tilde{\mathbf{f}}_j\|^2}{2\gamma^2}\right)$ ▷ Adjacency matrix
$\tilde{\mathbf{A}} \leftarrow \text{diag}(\mathbf{p})\mathbf{A}\text{diag}(\mathbf{p})$ ▷ Eq. (9)
$\mathbf{D} \leftarrow \text{diag}(d_1, \dots, d_N)$ where $d_i = \sum_j \tilde{A}_{ij}$ ▷ Degree matrix
$\mathbf{Q}_{struct} \leftarrow \mathbf{D} - \tilde{\mathbf{A}} + \alpha\mathbf{I}$ ▷ Eq. (10)
$\text{diag}(\boldsymbol{\Lambda}^{-1}) \leftarrow \text{ReLU}(\tanh(\tilde{\mathbf{F}}))$ ▷ Weight estimation
$\mathbf{H}_{\mathcal{G}} \leftarrow \mathbf{M}^T\boldsymbol{\Lambda}^{-1}\mathbf{Q}_{struct}\boldsymbol{\Lambda}^{-1}\mathbf{M}$
$\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{V} \leftarrow \text{SVD}(\mathbf{H}_{\mathcal{G}})$ ▷ Differentiable SVD
Extract $\mathbf{z}^*$ as the right singular vector corresponding to the smallest singular value (*i.e.*, the last column of $\mathbf{V}$).
return Estimated model parameter $\mathbf{z}^*$

4.2 Deep Topology Construction

According to the closed-form estimator, the key lies in the construction of $\mathbf{H}_{\mathcal{G}}$. Instead of relying on hand-crafted method, we make both $\mathbf{Q}_{struct}$ and $\boldsymbol{\Lambda}^{-1}$ learnable. As illustrated in Fig. 1, our proposed estimator is designed as a fully differentiable module that replaces WLS estimator at the end of a deep correspondence learning network (*e.g.*, OANet [32], DeMatch [33]). The upstream network processes the putative correspondences and outputs high-dimensional features $\mathbf{F} = \{\mathbf{f}_i\}_{i=1}^N$ and inlier probabilities $\mathbf{p} = \{p_i\}_{i=1}^N$ where $p_i \in [0, 1]$.

Our proposed method estimates the final geometric model parameters through the following sequential blocks:

Feature Embedding. We first pass the extracted features $\mathbf{F}$ through ResNet blocks [31] to obtain task-specific embeddings $\tilde{\mathbf{F}} = \{\tilde{\mathbf{f}}_i\}_{i=1}^N$.

Outlier-Filtered Adjacency Matrix. In order to compute the graph Laplacian matrix, we first construct a fully-connected adjacency matrix $\mathbf{A}$ via a learnable Gaussian kernel, $A_{ij} = \exp(-\|\tilde{\mathbf{f}}_i - \tilde{\mathbf{f}}_j\|^2/2\gamma^2)$, where γ is a learnable scalar. Notably, we model the GMRF only on inliers. If outliers are connected to inliers on the graph, the gross error of outliers will contaminate the inlier residuals via the graph Laplacian. To prevent this, the edges of outliers in the adjacency matrix must be strictly filtered as follows:

$$\tilde{\mathbf{A}} = \text{diag}(\mathbf{p})\mathbf{A}\text{diag}(\mathbf{p}), \tag{9}$$

which ensures that any edges connected to outliers (where $p_i \to 0$) is soft-pruned, effectively isolating outliers from the graph.

Precision Matrix. Given the outlier-filtered adjacency matrix $\tilde{\mathbf{A}}$, we compute its degree matrix $\mathbf{D} = \text{diag}(d_1, \dots, d_N)$ where $d_i = \sum_j \tilde{A}_{ij}$. To ensure the positive definite of the precision matrix, we employ the regularized graph Laplacian:

$$\mathbf{Q}_{struct} = \mathbf{D} - \tilde{\mathbf{A}} + \alpha\mathbf{I}, \tag{10}$$

where α demotes a small learnable regularization scalar and $\mathbf{I}$ denotes a learnable diagonal matrix.

Weight Estimation. In parallel with the graph construction, following [21], we compute $\boldsymbol{\Lambda}^{-1} = \text{diag}(1/\sigma_1, \ldots, 1/\sigma_N)$ through ReLU and Tanh layers.

Finally, following Eq. (7), the topological structure $\mathbf{Q}_{struct}$ and the individual weights $\boldsymbol{\Lambda}^{-1}$ are fused to form $\mathbf{Q}_{\mathcal{G}}$. This matrix is subsequently multiplied with the data matrix $\mathbf{M}$ to form the information matrix $\mathbf{H}_{\mathcal{G}}$, allowing the optimal geometric model parameters $\mathbf{z}^*$ to be efficiently computed via SVD layer. Furthermore, we summarize the overall algorithm of our structure-aware estimator in Algorithm 1.

5 Experiment

To evaluate the effectiveness and generalization ability of our proposed method, we conduct experiments across multiple tasks, including classic 2D line fitting, relative pose estimation, homography estimation, and long-term visual localization. Due to space limitations, more extended experiments and results are shown in the *supplementary material*.

5.1 Implementation Details

In the experiments, we evaluate the proposed structure-aware estimator by integrating it into various state-of-the-art (SOTA) correspondence learning networks [13, 16, 33–35] and reporting the performance gains. For instance, in relative pose estimation, we replace the original weighted eight-point algorithm [21] in the network with our estimator while keeping the remaining architectures unchanged, and then retrain the network following the default training strategy. Notably, the hyperparameters (*i.e.*, γ, α) of our estimator are complete learnable without any pre-defined parameters. All training and testing are performed on a single NVIDIA RTX5090 GPU.

5.2 Relative Pose Estimation

Relative pose estimation aims to recover the rotation and translation between two cameras observing the same scene from different viewpoints. Its accuracy therefore provides a direct indicator of how well an estimator computes the geometric model parameters.

Datasets. We conduct experiments on the outdoor YFCC100M [26] dataset. Putative correspondences are extracted using SIFT [15] with nearest-neighbor matching, and we keep up to 2000 matches per image pair. The keypoint coordinates are normalized using camera intrinsics. We use the majority of sequences for training/validation and reserve unseen sequences for cross-scene evaluation.

Evaluations. To demonstrate the effectiveness and lightweight nature of our estimator, experiments are conducted from two perspectives: computational resources and estimation accuracy. In terms of computational cost, the assessment

Table 1: Quantitative results of relative pose estimation on YFCC100M [26]. Accuracy improvements and computational efficiency drops are marked in red and blue.

Method	Computational Cost			Angular Error (AUC/mAP)		
	#Params (M) ↓	Time (ms) ↓	FLOPs (G) ↓	@5°↑	@10°↑	@20°↑
OANet [32]	2.473	4.375	1.841	15.97/39.27	35.92/53.77	57.09/67.67
CLNet [37]	0.954	7.036	1.921	20.34/43.65	38.79/55.12	57.80/67.94
NCMNet [14]	4.485	19.828	8.717	34.58/63.42	55.27/73.62	72.35/82.57
BCLNet [18]	4.460	16.219	10.063	35.70/65.88	56.59/75.30	73.15/83.41
ConvMatch [35]	7.494	8.843	3.783	26.78/57.05	49.10/68.81	67.87/78.82
ConvMatch*	7.611 +0.117	9.979 +1.136	4.022 +0.239	32.44/61.55 +5.66/+4.50	53.32/72.04 +4.22/+3.23	70.88/81.19 +3.01/+2.37
UMatch [13]	7.765	18.067	3.741	30.92/59.92	52.14/70.84	69.74/80.16
UMatch*	7.882 +0.117	19.327 +1.260	3.980 +0.239	34.37/64.38 +3.45/+4.46	55.16/73.75 +3.02/+2.91	72.04/82.32 +2.30/+2.16
DeMatch [33]	5.853	10.155	2.346	30.95/61.80	52.73/72.19	70.38/80.99
DeMatch*	5.970 +0.117	11.413 +1.258	2.586 +0.240	36.58/67.92 +5.63/+6.12	57.69/76.67 +4.96/+4.48	73.84/84.08 +3.46/+3.09
DeMo [16]	4.520	18.029	3.612	32.66/64.83	54.97/74.74	72.71/83.54
DeMo*	4.637 +0.117	18.897 +0.868	3.851 +0.239	38.01/69.80 +5.35/+4.97	59.48/78.67 +4.51/+3.93	75.86/86.31 +3.15/+2.77
Dematch++ [34]	5.986	13.753	3.744	34.33/65.92	56.00/75.43	72.93/83.43
Dematch++*	6.104 +0.118	15.029 +1.276	3.981 +0.237	40.03/71.15 +5.70/+5.23	60.67/79.04 +4.67/+3.61	76.13/86.04 +3.20/+2.61

covers key metrics including average inference time per image, floating-point operations (FLOPs), and the number of model parameters (#Params). In terms of estimation accuracy, following standard evaluation metrics, we report the area under the curve (AUC) of the pose error at thresholds of 5°, 10°, and 20°, as well as the mean average precision (mAP). We compare against representative learning-based methods [13, 14, 16, 18, 32–35, 37].

Results. The quantitative comparisons are listed in Tab. 1. It it clear that equipping existing networks with our proposed estimator (denoted as *) consistently yields substantial performance gains. In terms of computational overhead, the estimator introduces only a modest increase in parameter amount, inference time and FLOPs. We also discuss the limitation of our method in Sec. 5.7

5.3 Homography Estimation

Homography estimation aims to estimate a global planar transformation matrix in homogeneous coordinates. Compared with epipolar geometry, it provides a complementary evaluation of estimators under a different geometric model.

Datasets. We evaluate our method on the HPatches [2] benchmark, comprising 116 scenes featuring severe illumination and viewpoint variations. Each scene provides one reference image, five target images, and ground-truth homographies. We detect up to 4k SIFT keypoints per image and build putative correspondences via nearest-neighbor matching.

Evaluations. Following [9], an estimated homography is considered correct if the average corner reprojection error is below a threshold (*e.g.*, 3 pixels). We report the mean accuracy of the estimated homographies under different pixel-level error thresholds (1, 3, 5, 7, and 10 pixels).

Table 2: Results of homography estimation on the HPathes [2] dataset. Improvements and drops are marked in red and blue.

Method	@1px ↑	@3px ↑	@5px ↑	@7px ↑	@10px ↑
UMatch [13]	15.69	41.38	55.34	60.69	67.93
UMatch*	17.59 (+1.90)	49.48 (+8.10)	61.03 (+5.69)	67.76 (+7.07)	73.28 (+5.35)
ConvMatch [35]	20.00	47.41	61.03	66.72	73.62
ConvMatch*	18.97 (-1.03)	51.72 (+4.31)	63.10 (+2.07)	70.17 (+3.45)	75.86 (+2.24)
DeMatch [33]	21.55	44.83	55.69	63.97	71.55
DeMatch*	25.69 (+4.14)	48.45 (+3.62)	58.28 (+2.59)	64.83 (+0.86)	69.14 (-2.41)
DeMatch++ [34]	25.00	49.66	62.07	67.24	72.93
DeMatch++*	25.51 (+0.51)	55.34 (+5.68)	63.28 (+1.21)	68.97 (+1.73)	72.76 (-0.17)

Results. As reported in Tab. 2, networks equipped with our estimator surpass the baseline counterparts in most scenes. In particular, under the strict @3px threshold, incorporating our estimator into existing correspondence learning network achieves significant performance improvements. The results indicate that the proposed estimator improves the stability of the recovered transformation.

5.4 Visual Localization

Visual localization aims to estimate the 6-DoF camera pose of a query image with respect to a reference scene representation (*e.g.*, a 3D model), typically through 2D–3D correspondence establishment and geometric pose estimation.
Datasets. We conduct experiments on the Aachen Day-Night v1.1 [36]. This benchmark contains 6,697 reference images and a set of mobile-captured queries (824 day and 191 night). We extract up to 2,000 SIFT keypoints per image and build initial matches using mutual nearest neighbor matching.
Evaluations. We integrate correspondence learning networks into the HLoc [23] pipeline. COLMAP [24] is used to reconstruct the reference SfM model and recover query poses. All models are trained on YFCC100M with 2,000 SIFT keypoints. We report the percentage of localized queries under three commonly used thresholds: (0.25 m/2°), (0.5 m/5°), and (5.0 m/10°).
Results. Results are shown in Tab. 3. On daytime queries, the gains are relatively small, but it shows a significant improvement under nighttime conditions. This indicates that in poorly illuminated conditions, our estimator can effectively recover reliable geometric model parameters.

5.5 Line Fitting

The 2D line fitting, as one of the most fundamental geometric model estimation problems, can effectively reflect the performance of an estimator. Although the data are synthetic, it enables evaluation of an estimator under controlled setting with varying outlier ratios, noise levels, and occlusion percentages.
Datasets. We synthesize data by sampling a 2D line $ax + by + c = 0$ with (a, b, c) drawn randomly from $[0, 1]$. Inliers are generated by sampling $x \in [-1, 1]$ and computing y from the line equation (with Gaussian noise), while outliers are sampled uniformly in the square $[-1, 1] \times [-1, 1]$. Each instance contains

Table 3: Evaluation results of visual localization on the Aachen Day-Night v1.1 [36] dataset. Improvements and drops are marked in red and blue.

Method	Day ↑			Night ↑		
	0.25m, 2°	0.5m, 5°	5.0m, 10°	0.25m, 2°	0.5m, 5°	5.0m, 10°
RANSAC [10]	82.4	89.1	92.7	24.6	28.8	40.3
OANet [32]	85.4	93.2	96.5	45.0	55.5	74.3
UMatch [13]	85.7	92.6	96.6	50.8	62.3	77.5
UMatch*	85.4 (-0.3)	93.0 (+0.4)	97.3 (+0.7)	54.5 (+3.7)	66.5 (+4.2)	85.9 (+8.4)
ConvMatch [35]	85.6	93.3	96.5	51.8	64.9	81.2
ConvMatch*	85.4 (-0.2)	93.2 (-0.1)	97.0 (+0.5)	55.0 (+3.2)	68.1 (+3.2)	85.9 (+4.7)
DeMatch [33]	85.9	93.2	96.5	53.4	64.4	79.6
DeMatch*	86.0 (+0.1)	93.6 (+0.4)	97.1 (+0.6)	57.1 (+3.7)	69.6 (+5.2)	83.8 (+4.2)
DeMatch++ [34]	85.3	93.4	97.0	50.8	63.9	83.2
DeMatch++*	85.1 (-0.2)	93.2 (-0.2)	97.2 (+0.2)	56.5 (+5.7)	68.6 (+4.7)	85.9 (+2.7)

Table 4: L2 error computed by different methods on line fitting. Error decrease (*i.e.*, accuracy improvements) and increase (*i.e.*, accuracy drops) are marked in red and blue.

Method	Outlier Ratio				
	60%	70%	80%	90%	95%
OANet [32]	0.0254	0.0283	0.0360	0.0756	0.2720
OANet*	0.0260 (+0.0006)	0.0285 (+0.0002)	0.0346 (-0.0014)	0.0701 (-0.0055)	0.2381 (-0.0339)
UMatch [13]	0.0185	0.0219	0.0275	0.0476	0.1811
UMatch*	0.0175 (-0.0010)	0.0205 (-0.0014)	0.0265 (-0.0010)	0.0451 (-0.0025)	0.1559 (-0.0252)
Method	Occlusion Percentage				
	40%	50%	60%	70%	80%
OANet [32]	0.0345	0.0366	0.0416	0.0541	0.1131
OANet*	0.0339 (-0.0006)	0.0345 (-0.0021)	0.0397 (-0.0019)	0.0508 (-0.0033)	0.1043 (-0.0088)
UMatch [13]	0.0274	0.0296	0.0293	0.0377	0.0599
UMatch*	0.0239 (-0.0035)	0.0265 (-0.0031)	0.0276 (-0.0017)	0.0349 (-0.0028)	0.0560 (-0.0039)
Method	Noise Level				
	0.01	0.03	0.05	0.07	0.09
OANet [32]	0.0201	0.0300	0.0444	0.0602	0.0818
OANet*	0.0122 (-0.0079)	0.0253 (-0.0047)	0.0407 (-0.0037)	0.0602 (0.0000)	0.0792 (-0.0026)
UMatch [13]	0.0127	0.0190	0.0296	0.0416	0.0546
UMatch*	0.0097 (-0.0030)	0.0167 (-0.0023)	0.0278 (-0.0018)	0.0384 (-0.0032)	0.0527 (-0.0019)

$N = 1000$ points, and the goal is to recover (a, b, c). We use 10,000/2,000/2,000 instances for training/validation/testing. For comparison, we retrain OANet [32] and UMatch [13] using the official implementations.

Evaluations. We evaluated performance under three different test settings: an outlier ratio ranging from 50% to 90%, a noise standard deviation ranging from 0.01 to 0.1, and an occlusion percentage ranging from 40% to 80%. Following [37], in all experiments, we report the L2 error between the ground-truth (a, b, c) and the predicted ones.

Results. Table 4 summarizes the results of our line fitting experiment for different outlier ratio, occlusion percentage, and noise level settings. Except in scenarios with outlier ratios of 60% and 70%, where our method is slightly below the baseline, our proposed estimator achieves performance improvements in most other settings. Figure 2 visualizes the line fitting results of correspondence learning networks using the WLS estimator versus our proposed estimator, showing that our method fits lines closer to the ground truth.

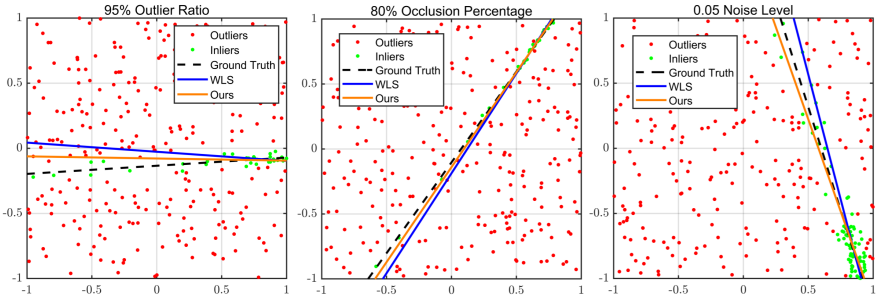

Fig. 2: Visualization results of OANet and OANet* on the 2D line fitting task.

Table 5: Generalization ability test reported by AUC@5°/mAP@5°. Improvements and drops are marked in red and blue.

Method	YFCC100M			SUN3D			
	RootSIFT	SuperPoint	Xfeat	SIFT	RootSIFT	SuperPoint	Xfeat
ConvMatch [35]	27.50/58.55	11.51/28.18	4.81/12.35	2.17/6.52	2.37/6.90	2.08/5.69	1.38/3.87
ConvMatch*	33.27/63.02	15.65/34.85	8.60/19.70	2.07/6.27	2.32/6.52	2.44/7.00	1.12/3.51
UMatch [13]	30.80/61.22	9.60/24.07	2.78/7.90	2.28/6.74	2.27/6.53	1.24/3.40	0.67/2.10
UMatch*	35.44/65.22	12.81/29.83	4.57/12.28	2.56/7.35	2.64/7.81	1.22/3.66	0.70/2.22
DeMatch [33]	31.23/61.60	11.48/28.27	6.03/14.65	2.21/6.50	2.17/6.42	1.88/5.08	0.98/2.83
DeMatch*	37.40/69.40	16.16/36.15	9.86/22.40	2.46/6.87	2.41/6.91	2.08/5.99	1.04/3.08
DeMo [16]	33.41/65.28	6.57/17.27	3.35/9.03	2.53/7.35	2.64/7.75	1.27/3.73	0.71/2.06
DeMo*	38.22/70.05	9.81/24.35	4.00/10.88	2.56/7.30	2.49/7.24	1.32/3.64	0.54/1.60
DeMatch++ [34]	34.95/67.03	13.57/31.35	7.33/17.27	2.64/7.58	2.68/7.71	1.65/4.71	1.18/3.11
DeMatch++*	40.66/71.47	16.31/36.52	10.24/23.28	2.73/7.75	2.74/7.98	1.37/4.07	0.54/1.70

5.6 Analysis

Generalization Ability. The performance of correspondence learning methods depends on both the initial descriptor quality and domain features. To assess the generalization ability, we train the networks solely on the outdoor YFCC100M using SIFT [15] descriptors, and test them directly on unseen descriptors (RootSIFT [1], SuperPoint [9], and XFeat [19]) and the unseen indoor SUN3D [29] dataset. All results are summarized in Tab. 5. Overall, integrating our estimator tends to bring consistent improvements across all descriptors on the training-domain dataset (YFCC100M). However, cross-domain generalization to SUN3D remains challenging: the improvements are unstable and sometimes even lower. This observation motivates our discussion of limitations in Sec. 5.7.

Ablation. To verify the effectiveness of each components in our estimator, we conduct ablation studies on YFCC100M dataset, using DeMatch [33] as the baseline. The results are summarized in Tab. 6. Experiment (i) denotes the DeMatch with the weighted eight-point [21] algorithm. Experiment (ii) refers to removing the graph Laplacian operation completely, which severely degrades performance, confirming the effectiveness of our design. In (iii) and (iv), we disable the degree matrix $\mathbf{D}$ and learnable regularization scalar α, respectively. Experiment (v) removes the weight matrix $\boldsymbol{\Lambda}^{-1}$, forcing the model to assume

Table 6: Ablation studies of our proposed estimator on the YFCC100M dataset using DeMatch [33] as the baseline.

Method	YFCC100M (AUC/mAP)		
	@5°	@10°	@20°
(i) Baseline	30.95/61.80	52.73/72.19	70.38/80.99
(ii) w/o. Lap	30.52/60.98	52.13/71.44	69.88/80.54
(iii) w/o. Degree	33.92/65.03	55.05/73.96	72.06/82.53
(iv) w/o. α	36.04/65.62	56.88/75.06	73.28/83.27
(v) w/o. $\boldsymbol{\Lambda}^{-1}$	36.21/66.38	56.96/75.54	73.26/83.40
(vi) Ours	36.58/67.92	57.69/76.67	73.84/84.08

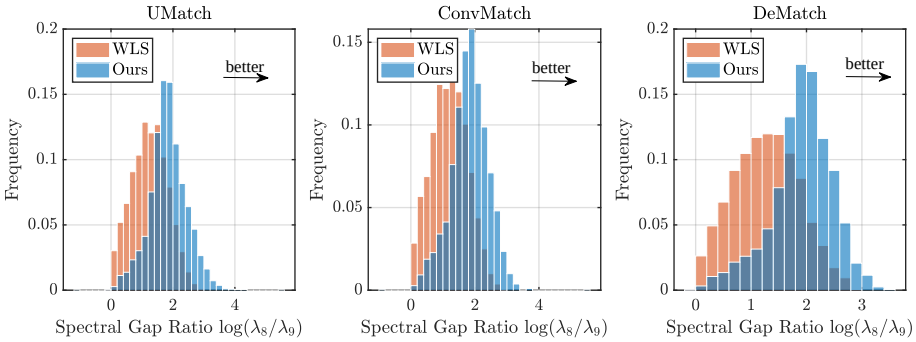

Fig. 3: The SGR histogram of information matrix $\mathbf{M}^T\mathbf{W}\mathbf{M}$ and $\mathbf{H}_{\mathcal{G}}$.

a uniform variance. Experiment (vi) is equipped with our complete estimator, achieving the optimal balance.

Effectiveness of the estimator. Our theoretical motivation comes from the observation that traditional WLS estimator yields a poorly-conditioned information matrix when inliers exhibit spatial clustering (non-uniform distribution), leading to unstable geometric model estimation. To validate that our estimator effectively mitigates this issue, we analyze the spectral properties of the information matrix used for the final SVD layer.

In geometric model estimation (*e.g.*, finding the fundamental matrix via the null-space of an information matrix $\mathbf{H}_{\mathcal{G}} \in \mathbb{R}^{9\times 9}$), a well-posed problem typically yields one distinct near-zero eigenvalue (λ_9) corresponding to the true model, while the second-smallest eigenvalue (λ_8) remains relatively larger. However, when inliers are spatially clustered (a near-degenerate configuration), their geometric constraints become highly linearly dependent. This redundancy drastically reduces λ_8, making the estimator highly susceptible to local noise.

To quantify this, we define a *spectral gap ratio* (SGR) metric as SGR $=$ $\log(\lambda_8/\lambda_9)$. A larger SGR indicates a more stable and unique solution space. We conduct experiments on the YFCC100M test set and summarize SGR histogram for both the traditional WLS matrix ($\mathbf{H}_{WLS} = \mathbf{M}^T\mathbf{W}\mathbf{M}$) and our information matrix ($\mathbf{H}_{\mathcal{G}}$). As shown in Fig. 3, across different network networks, our method consistently achieves a higher SGR, mathematically demonstrating that it effectively improves the spectral properties of the information matrix.

Table 7: Mean SGR of relevant ablations on the YFCC100M dataset using DeMatch [33] as the baseline.

Method	AUC/mAP @5°	Mean SGR
DeMatch	30.95/61.80	1.272
DeMatch* (ours)	36.58/67.92	1.907
DeMatch* w/o. Lap	30.52/60.98	1.283

Table 8: Results of alternative estimation strategies on YFCC100M dataset using DeMatch [33] as the baseline. AUC and mAP are reported.

Method	@5°	@10°	@20°
DeMatch	30.95/61.80	52.73/72.19	70.38/80.99
DeMatch+RANSAC [10]	32.55/59.62	51.67/69.12	68.45/78.32
DeMatch+MAGSAC++ [4]	34.18/62.88	54.38/72.56	71.00/81.03
DeMatch* (ours)	36.58/67.92	57.69/76.67	73.84/84.08

To better demonstrate the effectiveness of the graph Laplacian matrix, we also report mean SGR for relevant ablations in Tab. 7.

Alternative Estimation Strategies. Most existing correspondence learning networks primarily rely on the WLS-based estimator. Other types of robust estimators, like RANSAC [10], are commonly used as post-processing steps. We additionally report the performance of several robust estimators as post-processing in Tab. 8.

5.7 Limitations

Although our structure-aware estimator achieves strong performance, we identify two primary limitations in the current method that warrant future investigation:

Limited Performance on Pruning-based Architectures. We attempted to integrate our estimator into existing networks with the pruning operation (*e.g.*, CLNet [37] and BCLNet [18]). However, the training process becomes highly unstable, and the performance gain is very limited. We therefore report this limitation and provide a preliminary analysis. On the one hand, methods like BCLNet predict two vectors before executing the weighted eight-point algorithm: one is the inlier probability, and the other is the weight. This essentially means that their weighting strategy corresponds to making the Laplacian matrix a learnable diagonal matrix, which makes the estimation better than just predicting inlier probability. In these pruning-based methods, the improvement from adding the Laplacian matrix, compared to learning a diagonal weight matrix, is very limited (only about 1%). This might explain why the performance gain of pruning-based methods with our estimator is very limited. On the other hand, in [7], Dai et al. proposed that the pruning operation may reduce data abundance, and they proved it through experiments. From this perspective, we hypothesize that hard pruning operations may mistakenly remove a subset of true inliers, even if the remaining inliers are theoretically sufficient for minimal solvers (*e.g.*, 8 points for the fundamental matrix), the structural integrity required by our graph Laplacian matrix is compromised. Since GMRF inherently relies on the probabilistic

Table 9: Few-shot generalization ability test reported by AUC@5°/mAP@5°. Improvements and drops are marked in red and blue.

Method	YFCC100M			SUN3D			
	RootSIFT	SuperPoint	Xfeat	SIFT	RootSIFT	SuperPoint	Xfeat
ConvMatch [35]	27.88/58.38	10.47/25.95	4.43/11.15	2.49/7.22	2.63/7.51	1.98/5.54	1.27/3.52
ConvMatch*	33.92/64.40	15.15/33.40	6.32/15.12	3.53/10.01	3.73/10.52	2.80/8.15	1.64/4.88
UMatch [13]	30.81/60.92	8.21/21.95	1.86/5.30	2.82/8.28	2.95/8.11	1.37/3.97	0.74/2.17
UMatch*	39.40/69.60	13.07/28.43	4.48/11.47	4.04/10.89	4.10/11.30	1.80/4.95	0.89/2.78
DeMatch [33]	30.79/61.42	10.81/27.02	5.58/14.03	2.31/6.72	2.42/7.15	1.92/5.01	1.13/3.21
DeMatch*	37.30/67.92	15.84/34.60	9.51/20.72	3.18/9.06	3.28/9.45	2.05/5.79	1.07/3.31
DeMo [16]	32.83/64.68	6.58/17.40	3.20/8.45	2.46/7.31	2.63/7.40	1.18/3.36	0.75/2.14
DeMo*	39.55/71.10	11.19/26.17	4.07/9.98	4.30/11.78	4.43/12.12	1.31/3.68	0.43/1.43
DeMatch++ [34]	34.12/65.38	12.60/28.80	6.20/15.10	2.57/7.35	2.75/7.85	1.67/4.51	1.23/3.23
DeMatch++*	40.36/72.15	16.74/36.35	10.59/23.30	4.24/11.81	4.29/11.81	1.97/6.01	0.95/2.74

relationships among connected inliers to propagate structural constraints, we believe erroneous pruning may break the assumption of our method.

Cross-Dataset Generalization Gap. Although our method shows strong robustness across different descriptors, its generalization across fundamentally different geometric domains (*e.g.*, from outdoor YFCC100M to indoor SUN3D) is not yet satisfactory. We attribute this to the disparity in underlying physical manifolds: outdoor scenes typically feature distant and planar structures, whereas indoor environments consist of close-range and depth-varying surfaces. Consequently, the structural correlations learned by our estimator on outdoor data struggle to adapt to indoor topographies. To validate this analysis, we conducted a few-shot experiment freezing the feature backbone and fine-tuning only the estimation modules (both baseline WLS and our estimator) on a small subset of the SUN3D training data. As shown in Tab. 9, in this few-shot setting, our method demonstrates better generalization performance. We provide more results in the *supplementary material*. Future work will focus on enhancing the zero-shot generalization ability of our estimator.

6 Conclusion

In this paper, we addressed a fundamental bottleneck in current correspondence learning pipelines: the strict conditional independence assumption in WLS estimator limits the performance of model estimation accuracy. To bridge this gap, we introduced a novel structure-aware estimator. By explicitly modeling the probabilistic dependencies among inliers via the graph Laplacian matrix, our method effectively improves the information matrix used in SVD. As a lightweight and fully differentiable module, our estimator can be seamlessly integrated into most existing correspondence learning architectures, consistently yielding significant performance gains across multiple benchmarks with only a modest computational overhead.

References

1. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2911–2918. IEEE (2012)
2. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5173–5182 (2017)
3. Barath, D., Matas, J., Noskova, J.: Magsac: marginalizing sample consensus. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10197–10205 (2019)
4. Barath, D., Noskova, J., Ivashechkin, M., Matas, J.: Magsac++, a fast, reliable and accurate robust estimator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1304–1312 (2020)
5. Chum, O., Matas, J., Kittler, J.: Locally optimized ransac. In: Joint pattern recognition symposium. pp. 236–243. Springer (2003)
6. Dai, L., Du, X., Tang, J.: Trga: reconsidering the application of graph neural networks in two-view correspondence pruning. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 5633–5642 (2024)
7. Dai, L., Du, X., Zhang, H., Tang, J.: Mgnet: Learning correspondences via multiple graphs. In: Proceedings of the AAAI conference on Artificial Intelligence. vol. 38, pp. 3945–3953 (2024)
8. Dai, L., Liu, Y., Ma, J., Wei, L., Lai, T., Yang, C., Chen, R.: Ms2dg-net: Progressive correspondence learning via multiple sparse semantics dynamic graph. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8973–8982 (2022)
9. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 224–236 (2018)
10. Fischler, M.A., Bolles, R.C.: 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)
11. Green, P.J.: Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. Journal of the Royal Statistical Society: Series B (Methodological) **46**(2), 149–170 (1984)
12. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
13. Li, Z., Zhang, S., Ma, J.: U-match: Exploring hierarchy-aware local context for two-view correspondence learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
14. Liu, X., Yang, J.: Progressive neighbor consistency mining for correspondence pruning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9527–9537 (2023)
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004)
16. Lu, Y., Le, J., Li, Z., Yuan, Y., Ma, J.: Deep motion field consensus with learnable kernels for two-view correspondence learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 5829–5837 (2025)

17. Ma, J., Jiang, X., Fan, A., Jiang, J., Yan, J.: Image matching from handcrafted to deep features: A survey. International Journal of Computer Vision **129**(1), 23–79 (2021)
18. Miao, X., Xiao, G., Wang, S., Yu, J.: Bclnet: Bilateral consensus learning for two-view correspondence pruning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 4225–4232 (2024)
19. Potje, G., Cadar, F., Araujo, A., Martins, R., Nascimento, E.R.: Xfeat: Accelerated features for lightweight image matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2682–2691 (2024)
20. Raguram, R., Chum, O., Pollefeys, M., Matas, J., Frahm, J.M.: Usac: A universal framework for random sample consensus. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(8), 2022–2038 (2012)
21. Ranftl, R., Koltun, V.: Deep fundamental matrix estimation. In: Proceedings of the European conference on computer vision (ECCV). pp. 284–299 (2018)
22. Rue, H., Held, L.: Gaussian Markov random fields: theory and applications. Chapman and Hall/CRC (2005)
23. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: Robust hierarchical localization at large scale. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12716–12725 (2019)
24. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4104–4113 (2016)
25. Sun, W., Jiang, W., Trulls, E., Tagliasacchi, A., Yi, K.M.: Acne: Attentive context normalization for robust permutation-equivariant learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11286–11295 (2020)
26. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: Yfcc100m: The new data in multimedia research. Communications of the ACM **59**(2), 64–73 (2016)
27. Torr, P.H., Murray, D.W.: The development and comparison of robust methods for estimating the fundamental matrix. International Journal of Computer Vision **24**(3), 271–300 (1997)
28. Xiao, G., Liu, X., Zhong, Z., Zhang, X., Ma, J., Ling, H.: T-net++: Effective permutation-equivariance network for two-view correspondence pruning. IEEE transactions on pattern analysis and machine intelligence **46**(12), 10629–10644 (2024)
29. Xiao, J., Owens, A., Torralba, A.: Sun3d: A database of big spaces reconstructed using sfm and object labels. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1625–1632 (2013)
30. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: Learned invariant feature transform. In: Proceedings of the European Conference on Computer Vision. pp. 467–483. Springer (2016)
31. Yi, K.M., Trulls, E., Ono, Y., Lepetit, V., Salzmann, M., Fua, P.: Learning to find good correspondences. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2666–2674 (2018)
32. Zhang, J., Sun, D., Luo, Z., Yao, A., Zhou, L., Shen, T., Chen, Y., Quan, L., Liao, H.: Learning two-view correspondences and geometry using order-aware network. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5845–5854 (2019)

33. Zhang, S., Li, Z., Gao, Y., Ma, J.: Dematch: Deep decomposition of motion field for two-view correspondence learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 20278–20287 (June 2024)
34. Zhang, S., Li, Z., Ma, J.: Dematch++: Two-view correspondence learning via deep motion field decomposition and respective local-context aggregation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
35. Zhang, S., Ma, J.: Convmatch: Rethinking network design for two-view correspondence learning. In: Proceedings of the AAAI conference on Artificial Intelligence. vol. 37, pp. 3472–3479 (2023)
36. Zhang, Z., Sattler, T., Scaramuzza, D.: Reference pose generation for long-term visual localization via learned features and view synthesis. International Journal of Computer Vision **129**(4), 821–844 (2021)
37. Zhao, C., Ge, Y., Zhu, F., Zhao, R., Li, H., Salzmann, M.: Progressive correspondence pruning by consensus learning. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6464–6473 (2021)