

# End2End Multi-View Feature Matching with Differentiable Pose Optimization

Barbara Roessle and Matthias Nießner  
 Technical University of Munich

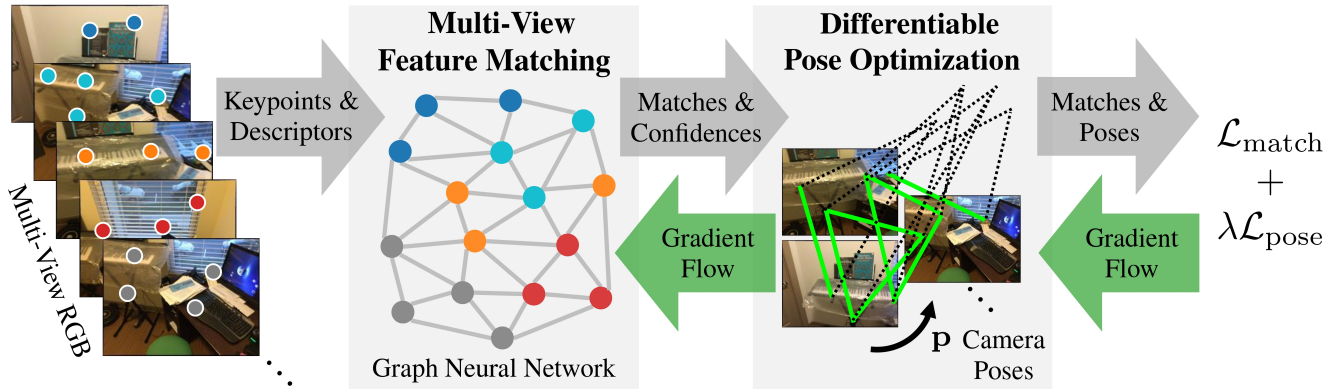


Figure 1. We connect feature matching and pose optimization in an end-to-end trainable approach that enables matches and confidence weights to be informed by the pose estimation objective. To this end, we introduce GNN-based multi-view feature matching to predict matches and confidences tailored to a differentiable pose solver, which significantly improves pose estimation performance.

## Abstract

*Erroneous feature matches have severe impact on subsequent camera pose estimation and often require additional, time-costly measures, like RANSAC, for outlier rejection. Our method tackles this challenge by addressing feature matching and pose optimization jointly. To this end, we propose a graph attention network to predict image correspondences along with confidence weights. The resulting matches serve as weighted constraints in a differentiable pose estimation. Training feature matching with gradients from pose optimization naturally learns to down-weight outliers and boosts pose estimation on image pairs compared to SuperGlue by 6.7% on ScanNet. At the same time, it reduces the pose estimation time by over 50% and renders RANSAC iterations unnecessary. Moreover, we integrate information from multiple views by spanning the graph across multiple frames to predict the matches all at once. Multi-view matching combined with end-to-end training improves the pose estimation metrics on Matterport3D by 18.5% compared to SuperGlue.*

## 1. Introduction

Feature matching is a key component in many 3D vision applications such as structure from motion (SfM) or simultaneous localization and mapping (SLAM). Conventional pose estimation is a multi-step process: feature detection finds interest points, for which local descriptors are computed. Based on the descriptors, pairs of keypoints from different images are matched, which defines constraints in the pose optimization. A major challenge lies in the ambiguity of matching local descriptors by nearest-neighbor search, which is error-prone, particularly in texture-less areas or in presence of repetitive patterns. Hand-crafted heuristics or outlier filters become necessary to circumvent this problem to some degree.

Recent learning-based approaches [45, 48, 26, 35] instead leverage the greater image context to improve the matching, e.g., SuperGlue [45] introduces a graph neural network (GNN) for descriptor matching on an image pair. Graph edges connect keypoints from arbitrary locations and enable reasoning in a broad context, leading to globally well-informed solutions compared to convolutional neural networks (CNN) with limited receptive field. The receptive field in SuperGlue, however, remains limited by the two-

view setup, despite that more images are typically available in pose estimation tasks. Our idea is to further facilitate information flow by joining multiple views in the matching process. This way, we allow multi-view correlation to strengthen geometric reasoning and confidence prediction. Joint matching of multiple images integrates well into pose estimation pipelines, as they typically solve for more than two cameras.

Additionally, we note that accurate feature matching, in and of itself, does not necessarily give rise to accurate pose estimation, as the spatial distribution of feature matches is essential for robust pose optimization. For instance, perfectly precise matches may form a degenerate case (e.g., lying on a line) and thus have no value for pose optimization. In addition, confidence scores predicted by matching networks do not necessarily reflect the value of matches towards pose optimization. Feature matching and pose estimation are thus tightly coupled problems, for which we propose a joint solution.

We encode keypoints and descriptors from multiple images to construct a graph, where self-attention provides context awareness within the same image and cross-attention enables reasoning with respect to all other images. A GNN predicts matches along with confidence weights, which define constraints on the camera poses that we optimize with a differentiable solver. The GNN is trained end-to-end using gradients from the pose optimization. From this feedback, the network learns to produce valuable matches for pose estimation and thereby learns effective outlier rejection. We evaluate our method on image pairs and in a multi-view setting on ScanNet [14], Matterport3D [10], and MegaDepth [30] datasets and show that our joint approach to feature matching and pose estimation improves over prior work on learned feature matching, enabled by the following contributions:

- We introduce an end-to-end trainable pose estimation that both guides confidence weights of feature matches in an unsupervised fashion and backpropagates gradients to inform the matching network.
- We propose a multi-view graph attention network to learn feature matches simultaneously across multiple frames.

## 2. Related Work

**Conventional Feature Matching.** The classical feature matching pipeline comprises the following steps: 1) interest point detection, 2) feature description, 3) matching through nearest neighbor search in descriptor space, and 4) outlier filtering. In this pipeline, hand-crafted features like SIFT [32] and ORB [44] are very successful and have been widely used for many years. However, they tend

to struggle with appearance or viewpoint changes. Starting with LIFT [55], learning-based descriptors have been developed to tackle these challenges [36, 17, 41, 4, 52]. They often combine interest point detection and description, such as SuperPoint [16], which we use for our method. Nearest neighbor feature matching is prone to outliers, making post-processing methods indispensable. This includes mutual check, ratio test [32], neighborhood consensus [51, 9, 8, 5, 34] and sampling-based outlier rejection [19, 3, 39]. Learning-based approaches have also addressed outlier detection [56, 40, 7, 58]—these methods rely on reasonable matching proposals and lack visual information in their decision process.

**Learning Feature Matching.** Recent methods employ neural networks for feature matching on image pairs. There are methods that determine dense, pixel-wise correspondences with confidence estimates for filtering [43, 42, 29]. However, the matching lacks global context due to the limited receptive field of CNNs and fails to distinguish regions of little texture or repetitive structure. In contrast, SuperGlue [45] represents a sparse matching network that operates on keypoints with descriptors. Using an attentional GNN [54] all keypoints interact, hence the receptive field spans across both images, leading to accurate matches in wide-baseline settings. Inspired by GNN-based feature matching, we build upon SuperGlue by enhancing its receptive field through multi-view matching and by improving outlier filtering through end-to-end training with pose optimization. LoFTR [48] and COTR [26] recently proposed detector-free methods that operate on RGB images directly. Using attention and a coarse-to-fine approach, they equally achieve a receptive field across the image pair and high quality matches. 3DG-STFM [35] extends LoFTR with student-teacher learning to leverage RGB-comprised depth information. We show that our end-to-end and multi-view approach improves pose estimation over SuperGlue and the detector-free methods LoFTR, COTR, and 3DG-STFM.

**Pose Optimization.** Once matches between a set of images are found, bundle adjustment formulations [50] are used to optimize poses on RGB [1] or RGB-D data [15]. This typically leads to non-linear least squares problems which are optimized with non-linear solvers, like Gauss-Newton or Levenberg-Marquardt. Such pipelines usually perform feature matching as a pre-process, followed by a filtering with a combination of RANSAC and robust optimization techniques [57, 12]. However, feature matching and pose optimization largely remain separate steps and cannot inform each other. To this end, differentiable pose optimization techniques, such as DeMoN [53], BANet [49], RegNet [22], or 3DRegNet [38], propose to obtain gradients through the pose optimization that in turn guide the learning of feature descriptors. In contrast to treating feature extraction as a separate step, feature descriptors are

then learned with the objective to obtain well-aligned poses. In our work, we go a step further and focus on learning how to match features rather than using a predefined matching method. We leverage differentiable pose optimization to provide gradients for our feature matching network, and achieve significantly improved pose estimation results.

### 3. Method

Our method associates keypoints from  $N$  images  $\{I_n\}_{n=1}^N$ , such that the resulting matches and confidence weights are particularly valuable for estimating the corresponding camera poses  $\{\mathbf{p}_n\}_{n=1}^N$ ;  $\mathbf{p}_n \in \mathbb{R}^6$ . Keypoints are represented by their image coordinates  $\mathbf{x} \in \mathbb{R}^2$ , visual descriptors  $\mathbf{d} \in \mathbb{R}^D$  and a confidence score  $c \in [0, 1]$ . We use the SuperPoint network for feature detection and description [16]. Our pipeline (Fig. 1) ties together feature matching and pose optimization: we employ a GNN to associate keypoints across multiple images (Sec. 3.1). The resulting matches and confidence weights define constraints in the subsequent pose optimization (Sec. 3.2), which is differentiable, thus enabling end-to-end training (Sec. 3.3). Both, multi-view and end-to-end, are independent and can be used in isolation, however, the benefit is larger in combination, as shown in the experiments (Sec. 4).

#### 3.1. Multi-View Graph Attention Network

**Motivation.** In the multi-view matching problem of  $N$  images, each keypoint matches to at most  $N - 1$  other keypoints, where each of the matching keypoints belongs to a different input image. Without knowing the transformations between images, one keypoint can match to any keypoint location in the other images. Hence, all keypoints in the other images need to be considered as matching candidates. Although keypoints from the same image are not matching candidates, they contribute valuable constraints in the assignment problem, e.g., their projection into other images must follow consistent transformations. The matching problem can be represented as a graph, where nodes model keypoints and edges their relationships. A GNN architecture reflects this structure and enables learning the complex relations between keypoints to determine feature matches. The iterative message passing process enables the search for globally optimal matches as opposed to a greedy local assignment. On top of that, attention-based message aggregation allows each keypoint to focus on information from the keypoints that provide the most insight for its assignment. We build upon SuperGlue, which introduces an attentional GNN for descriptor matching on image pairs [45]. Our extension to multi-image matching is motivated by the following: first, graph-based reasoning can benefit from tracks that are longer than two keypoints—i.e., a match becomes more confident, if multiple views agree on the keypoint similarity and its coherent location with respect to the other key-

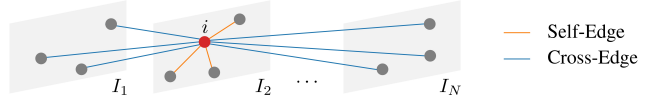


Figure 2. Keypoints are graph nodes. Keypoint  $i$  is connected to keypoints in the same image through self-edges and to keypoints in other images through cross-edges.

points in each frame. In particular, with regards to robust pose optimization, it is crucial to facilitate this information flow and boost the confidence prediction. Second, pose estimation or SLAM systems generally consider multiple input views. With the described graph structure, jointly matching  $N$  images is more efficient in terms of intra-frame GNN messages than matching the corresponding image pairs individually, as detailed in the supplementary material.

**Graph Construction.** Each keypoint represents a graph node. The initial node embedding  ${}^{(1)}\mathbf{f}_i$  of keypoint  $i$  is computed from its image coordinate  $\mathbf{x}_i$ , confidence  $c_i$  and descriptor  $\mathbf{d}_i$ , which allows the GNN to consider spatial location, certainty and visual appearance in the matching:

$${}^{(1)}\mathbf{f}_i = \mathbf{d}_i + F_{\text{encode}}([\mathbf{x}_i \parallel c_i]), \quad (1)$$

where  $\parallel$  denotes row-wise concatenation.  $F_{\text{encode}}$  is a multilayer perceptron (MLP) that lifts the image point and its confidence into the high-dimensional space of the descriptor to help the spatial learning [45, 20, 54]. The graph nodes are connected by two kinds of edges: self-edges connect keypoints within the same image. Cross-edges connect keypoints from different images (Fig. 2). The edges are undirected, i.e., information flows in both directions.

**Message Passing.** Interaction between keypoints—the graph nodes—is realized through message passing [18, 21]. The goal is to achieve a state where node descriptors of matching keypoints are close in descriptor space, whereas unrelated keypoints are far apart. The GNN has  $L$  layers, where each layer  $\ell$  corresponds to a message exchange between keypoints. The layers alternate between updates along self-edges  $\mathcal{E}_{\text{self}}$  and cross-edges  $\mathcal{E}_{\text{cross}}$ —starting with an exchange along self-edges in layer  $\ell = 1$  [45]. Eq. (2) describes the iterative node descriptor update, where  ${}^{(\ell)}\mathbf{m}_{\mathcal{E} \rightarrow i}$  is the aggregated message from all keypoints that are connected to keypoint  $i$  by an edge in  $\mathcal{E} \in \{\mathcal{E}_{\text{self}}, \mathcal{E}_{\text{cross}}\}$ .  ${}^{(\ell)}F_{\text{update}}$  is a MLP, where each GNN layer  $\ell$  has a separate set of network weights.

$${}^{(\ell+1)}\mathbf{f}_i = {}^{(\ell)}\mathbf{f}_i + {}^{(\ell)}F_{\text{update}}\left(\left[{}^{(\ell)}\mathbf{f}_i \parallel {}^{(\ell)}\mathbf{m}_{\mathcal{E} \rightarrow i}\right]\right) \quad (2)$$

Multi-head attention [54] is used to merge all incoming information for keypoint  $i$  into a single message  ${}^{(\ell)}\mathbf{m}_{\mathcal{E} \rightarrow i}$  [45]. Messages along self-edges are combined by self-attention between the keypoints of the same image, messages along cross-edges by cross-attention between the

keypoints from all other images. Linear projection of node descriptors is used to compute the query  ${}^{(\ell)}\mathbf{q}_i$  of query keypoint  $i$ , as well as the keys  ${}^{(\ell)}\mathbf{k}_j$  and values  ${}^{(\ell)}\mathbf{v}_j$  of its source keypoints  $j$ :

$${}^{(\ell)}\mathbf{q}_i = {}^{(\ell)}\mathbf{W}_1 {}^{(\ell)}\mathbf{f}_i + {}^{(\ell)}\mathbf{b}_1, \quad (3)$$

$$\begin{bmatrix} {}^{(\ell)}\mathbf{k}_j \\ {}^{(\ell)}\mathbf{v}_j \end{bmatrix} = \begin{bmatrix} {}^{(\ell)}\mathbf{W}_2 \\ {}^{(\ell)}\mathbf{W}_3 \end{bmatrix} {}^{(\ell)}\mathbf{f}_j + \begin{bmatrix} {}^{(\ell)}\mathbf{b}_2 \\ {}^{(\ell)}\mathbf{b}_3 \end{bmatrix}. \quad (4)$$

The set of source keypoints  $\{j : (i, j) \in \mathcal{E}\}$  comprises all keypoints connected to  $i$  by an edge of the type, that is relevant to the current layer.  $\mathbf{W}$  and  $\mathbf{b}$  are per-layer weight matrices and bias vectors, respectively. For each source keypoint the similarity to the query is computed by the dot product  ${}^{(\ell)}\mathbf{q}_i \cdot {}^{(\ell)}\mathbf{k}_j$ . The softmax over the similarity scores determines the attention weight  $\alpha_{ij}$  of each source keypoint  $j$  in the aggregated message to  $i$ :

$${}^{(\ell)}\mathbf{m}_{\mathcal{E} \rightarrow i} = \sum_{j:(i,j) \in \mathcal{E}} {}^{(\ell)}\alpha_{ij} {}^{(\ell)}\mathbf{v}_j. \quad (5)$$

It is important to note that in cross-attention layers, the source keypoints  $j$  to a query keypoint  $i$  come from multiple images. The softmax-based weighting is robust to variable number of input views and therewith variable number of keypoints. After  $L$  message passing iterations the node descriptors for subsequent assignment are retrieved by linear projection:

$$\mathbf{f}_i = \mathbf{W}_4 {}^{(L+1)}\mathbf{f}_i + \mathbf{b}_4. \quad (6)$$

**Partial Assignment.** The partial assignment problem of keypoints from two images can be solved with the differentiable Sinkhorn algorithm [47, 13, 45]: Given an input score matrix, a partial assignment is optimized, where each keypoint either obtains a match in the other image or remains unmatched. We compute the assignment on the set of possible image pairs  $\mathcal{P}$ , excluding pairs between identical images and pairs that are a permutation of another pair. For each pair  $(a, b) \in \mathcal{P}; a, b \in \{1, 2, \dots, N\}$ , the score matrix is filled with the dot-product similarities of node descriptors. From the resulting partial assignment matrix  $\mathbf{P}_{ab}$ , the set of matches is derived: first, a candidate match for each keypoint is determined by the row-wise and column-wise maximal elements. Second, we keep only those matches, where both keypoints mutually agree on the assignment.

**Confidence Prediction.** For each pair of matching keypoints  $i, j$  a confidence weight  $w_{ij}$  is predicted from the final node descriptors  $\mathbf{f}_i, \mathbf{f}_j$  and their score in the corresponding partial assignment matrix  $\mathbf{P}_{ab}$ :

$$w_{ij} = F_{\text{conf}_1}(F_{\text{conf}_2}(\mathbf{P}_{ab, i, j}) + F_{\text{conf}_3}([\mathbf{f}_i \parallel \mathbf{f}_j])), \quad (7)$$

where  $F_{\text{conf}_*}$  represent small MLPs.

## 3.2. Differentiable Pose Optimization

We introduce a differentiable relative pose optimization that provides supervision signal for feature matching. It is composed of two parts: initial pose estimation through a weighted eight-point algorithm and pose refinement through bundle adjustment.

**Weighted Eight-Point Algorithm.** For each image pair, a fundamental matrix  $\mathbf{F}$  is computed using the eight-point algorithm [31] with input coordinate normalization [23]. To facilitate the learning of meaningful confidences, it is essential to consider all matches in a weighted manner. Hence, we define the system of linear equations as a confidence-weighted version of the eight-point algorithm:

$$\text{diag}(\mathbf{w})\mathbf{A} \text{flat}(\mathbf{F}) = \mathbf{0}. \quad (8)$$

Eq. (8) follows from the epipolar geometry  $\mathbf{x}'^\top \mathbf{F} \mathbf{x} = 0$  by arranging the known coordinates of a match,  $\mathbf{x} = [x, y, z]^\top$  and  $\mathbf{x}' = [x', y', z']^\top$ , into matrix  $\mathbf{A}$  and flattening  $\mathbf{F}$  in column-major order to a vector  $\text{flat}(\mathbf{F})$ . Each row  $[xx', xy', x, yx', yy', y, x', y', 1]$  in  $\mathbf{A}$  describes one match and is multiplied with its confidence through the diagonal matrix  $\text{diag}(\mathbf{w})$  from the vector of confidences  $\mathbf{w}$ . Given more than 8 matches, the system is overdetermined. Thus, we search a least-squares solution for  $\mathbf{F}$  that minimizes  $\|\text{diag}(\mathbf{w})\mathbf{A} \text{flat}(\mathbf{F})\|_2$  under the constraint  $\|\text{flat}(\mathbf{F})\|_2 = 1$  to avoid the trivial solution. Singular value decomposition (SVD) of  $\text{diag}(\mathbf{w})\mathbf{A}$  determines this solution as the singular vector with the smallest singular value and we force the resulting  $\mathbf{F}$  to have rank 2 [24]. The partial derivatives of the SVD can be computed in closed-form [25], thus the eight-point algorithm suits well for end-to-end training. Given the intrinsics and the resulting  $\mathbf{F}$ , there are four possible solutions for the relative transformation between an image pair, aside from unknown scale. During training, we select the solution closest to the ground truth. At test time, following the cheirality constraint [24], the solution with most triangulated points in front of both cameras is chosen.

**Bundle Adjustment.** The initial relative pose  $\mathbf{p}_{\text{init}}$  from the weighted eight-point algorithm is refined using a bundle adjustment formulation. To this end, we introduce a differentiable optimizer  $\Omega$  to refine the relative pose  $\mathbf{p}$  and estimate 3D points  $\mathbf{Y} \in \mathbb{R}^{M \times 3}$  for the matches  $\mathcal{M}$ :

$$\{\mathbf{p}, \mathbf{Y}\} = \Omega(\mathbf{p}_{\text{init}}, \mathcal{M}). \quad (9)$$

For each match  $m$ , we compute confidence-weighted residuals  $\mathbf{r}_m, \mathbf{r}'_m \in \mathbb{R}^2$  on the projection of the corresponding 3D point  $\mathbf{y}$  into each image and define the energy as the sum of squares:

$$E(\mathbf{p}, \mathbf{Y}) = \sum_{(\mathbf{x}, \mathbf{x}', w), \mathbf{y} \in \mathcal{M}, \mathbf{Y}} \left( \|\mathbf{r}_m\|_2^2 + \|\mathbf{r}'_m\|_2^2 \right), \quad \text{where} \quad (10)$$

$$\mathbf{r}_m = w(\pi(\mathbf{y}) - \mathbf{x}), \quad \mathbf{r}'_m = w(\pi'(\mathbf{R}\mathbf{y} + \mathbf{t}) - \mathbf{x}'). \quad (11)$$



$\mathbf{x}$  and  $\mathbf{x}'$  are the image coordinates of a match and  $w$  is its confidence. The 3D points are defined in the first camera frame and  $\{\mathbf{R} \in \mathbb{R}^{3 \times 3}, \mathbf{t} \in \mathbb{R}^3\}$  describes the transformation from the first to the second camera, for which  $\mathbf{p} \in \mathbb{R}^6$  is the equivalent pose vector in  $\mathfrak{se}(3)$  coordinates, i.e., three translation elements followed by three rotation elements. The functions,  $\pi$  and  $\pi'$ , project a 3D point from the respective camera frame to its image plane.  $\mathbf{p}$  is initialized to  $\mathbf{p}_{\text{init}}$  and  $\mathbf{Y}$  is initialized by triangulating the matches.

Gauss-Newton algorithm is used to minimize the energy with respect to the relative pose and the 3D points. Thus, we optimize for a vector  $\mathbf{z} = [\mathbf{p} \parallel \text{flat}(\mathbf{Y}^\top)] \in \mathbb{R}^{6+3M}$  and compose a residual vector  $\mathbf{r} = [\mathbf{r}_1 \parallel \mathbf{r}'_1 \parallel \dots \parallel \mathbf{r}_M \parallel \mathbf{r}'_M] \in \mathbb{R}^{4M}$ , where  $M$  is the number of matches. The Jacobian matrix  $\mathbf{J} \in \mathbb{R}^{4M \times (6+3M)}$  is initialized to  $\mathbf{0}$  and for each match  $m$  the corresponding submatrices are filled with the partial derivatives with respect to the pose  $\frac{\partial \mathbf{r}'_m}{\partial \mathbf{p}} \in \mathbb{R}^{2 \times 6}$  and with respect to the 3D point  $\frac{\partial \mathbf{r}_m}{\partial \mathbf{y}}, \frac{\partial \mathbf{r}'_m}{\partial \mathbf{y}} \in \mathbb{R}^{2 \times 3}$  [6]:

$$\frac{\partial \mathbf{r}'_m}{\partial \mathbf{p}} = w \frac{\partial \pi'(\mathbf{R}\mathbf{y} + \mathbf{t})}{\partial (\mathbf{R}\mathbf{y} + \mathbf{t})} [\mathbf{I} \quad -(\mathbf{R}\mathbf{y} + \mathbf{t})^\wedge], \quad (12)$$

$$\frac{\partial \mathbf{r}_m}{\partial \mathbf{y}} = w \frac{\partial \pi(\mathbf{y})}{\partial \mathbf{y}}, \quad \frac{\partial \mathbf{r}'_m}{\partial \mathbf{y}} = w \frac{\partial \pi'(\mathbf{R}\mathbf{y} + \mathbf{t})}{\partial (\mathbf{R}\mathbf{y} + \mathbf{t})} \mathbf{R}, \quad (13)$$

$$\text{where } \frac{\partial \pi(\mathbf{u})}{\partial \mathbf{u}} = \begin{bmatrix} f_x/u_z & 0 & -f_x u_x/u_z^2 \\ 0 & f_y/u_z & -f_y u_y/u_z^2 \end{bmatrix}. \quad (14)$$

$\mathbf{I}$  is a  $3 \times 3$  identity matrix,  $(\cdot)^\wedge$  maps a vector  $\in \mathbb{R}^3$  to its skew-symmetric matrix,  $f_*$  are focal lengths and  $u_*$  are coordinates of a 3D point  $\mathbf{u}$ .

Using the current state of  $\{\mathbf{p}, \mathbf{Y}\}$ , each Gauss-Newton iteration establishes a linear system, that is solved for the update  $\Delta \mathbf{z}$  using LU decomposition:

$$\mathbf{J}^\top \mathbf{J} \Delta \mathbf{z} = -\mathbf{J}^\top \mathbf{r}. \quad (15)$$

We update the state in  $T$  Gauss-Newton iterations and apply Jacobi preconditioning and a damping factor  $\beta$  for stability.

### 3.3. End-to-End Training

The whole pipeline, from the matching network to the pose optimization, is differentiable, which allows for a pose loss that guides the matching network to produce valuable matches and accurate confidences for robust pose optimization. The training objective  $\mathcal{L}$  consists of a matching term  $\mathcal{L}_{\text{match}}$  [45] and a pose term  $\mathcal{L}_{\text{pose}}$ , which are balanced by

the factor  $\lambda$ :

$$\mathcal{L} = \sum_{(a,b) \in \mathcal{P}} \mathcal{L}_{\text{match}}(a,b) + \lambda \mathcal{L}_{\text{pose}}(a,b), \quad \text{where} \quad (16)$$

$$\mathcal{L}_{\text{match}}(a,b) = - \sum_{(i,j) \in \mathcal{T}_{ab}} \log \mathbf{P}_{ab,i,j} \quad (17)$$

$$- \sum_{i \in \mathcal{U}_{ab}} \log \mathbf{P}_{ab,i,j_{\text{max}}} - \sum_{j \in \mathcal{V}_{ab}} \log \mathbf{P}_{ab,i_{\text{max}},j},$$

unmatched                      unmatched

$$\mathcal{L}_{\text{pose}}(a,b) = \cos^{-1} \left( \frac{\hat{\mathbf{t}}_{a \rightarrow b} \cdot \mathbf{t}_{a \rightarrow b}}{\|\hat{\mathbf{t}}_{a \rightarrow b}\|_2 \cdot \|\mathbf{t}_{a \rightarrow b}\|_2} \right) \quad (18)$$

$$+ \lambda_{\text{rot}} \cos^{-1} \left( \frac{\text{tr}(\hat{\mathbf{R}}_{a \rightarrow b}^\top \mathbf{R}_{a \rightarrow b}) - 1}{2} \right).$$

$\mathcal{L}_{\text{match}}$  computes the negative log-likelihood of the assignment between an image pair. The labels are computed using the ground truth depth maps and camera parameters:  $\mathcal{T}_{ab}$  is the set of matching keypoints,  $\mathcal{U}_{ab}$  and  $\mathcal{V}_{ab}$  identify unmatched keypoints from  $I_a$  and  $I_b$ , respectively.  $\mathcal{L}_{\text{pose}}$  computes a transformation error between a pair of camera poses, where the translational and rotational components are balanced by  $\lambda_{\text{rot}}$ . We found that training on the weighted eight-point result works equally well as training on both weighted eight-point and bundle adjustment, hence,  $\mathcal{L}_{\text{pose}}$  is applied on the weighted eight-point result. At test time, however, the pose refinement with bundle adjustment is highly beneficial as shown in the experiments (Sec. 4).  $\hat{\mathbf{R}}_{a \rightarrow b}$  and  $\hat{\mathbf{t}}_{a \rightarrow b}$  are the rotation matrix and translation vector of the estimated pose.  $\mathbf{R}_{a \rightarrow b}$  and  $\mathbf{t}_{a \rightarrow b}$  define the ground truth transformation. We use the Adam optimizer [28]. Further detail on the network architecture and training setup are provided in the supplementary material.

## 4. Results

We evaluate performance on indoor and outdoor pose estimation in a two-view and multi-view setting (Secs. 4.1 and 4.2) and runtime (Sec. 4.3). Sec. 4.4 shows the effectiveness of end-to-end training and multi-view matching in an ablation study. A cross-dataset and matching evaluation is provided in the supplement.

**Baselines.** Prior work, in particular SuperGlue [45], has extensively demonstrated the superiority of the GNN approach over conventional matching. Hence, we focus on comparisons to recent matching networks: SuperGlue [45], LoFTR [48], COTR [26], and 3DG-STFM [35]. We additionally compare to a non-learning-based matcher, i.e., mutual nearest neighbor search on the SuperPoint [16] descriptors. This serves to confirm the effectiveness of SuperGlue and our method, which both use SuperPoint descriptors.

### 4.1. Two-View Pose Estimation

Following prior work [45, 48, 35], we evaluate on the same 1500 image pairs of ScanNet and MegaDepth and

	Pose est. method	Pose error AUC [%] $\uparrow$		
		@5°	@10°	@20°
Mutual nearest neighbor	RANSAC	9.5	21.6	35.7
SuperGlue [45]		16.2	33.8	51.8
LoFTR [48]		22.1	40.8	57.6
COTR [26] cross-dataset		11.8	26.5	42.5
3DG-STFM [35]		23.6	43.6	61.2
Ours w/o multi-view		20.7	41.3	60.7
Mutual nearest neighbor	Weight. 8-point	0.0	0.1	0.7
SuperGlue [45]		11.7	26.8	45.6
LoFTR [48]		15.0	30.6	47.3
COTR [26] cross-dataset		3.2	9.5	20.2
3DG-STFM [35]		10.1	23.4	39.5
Ours w/o multi-view		20.7	41.6	61.7
Mutual nearest neighbor	RANSAC + bundle adjust.	10.1	22.4	36.3
SuperGlue [45]		17.0	35.2	54.0
LoFTR [48]		22.4	41.0	57.7
COTR [26] cross-dataset		12.6	27.7	43.5
3DG-STFM [35]		23.3	42.4	59.1
Ours w/o multi-view		23.1	43.6	62.3
Mutual nearest neighbor	Weight. 8-point + bundle adjust.	0.0	0.3	1.8
SuperGlue [45]		20.6	40.0	58.7
LoFTR [48]		24.0	42.8	59.1
COTR [26] cross-dataset		8.5	19.6	33.9
3DG-STFM [35]		20.3	37.9	54.1
Ours w/o multi-view		<b>25.7</b>	<b>47.2</b>	<b>66.4</b>

Table 1. Baseline comparison on two-view, wide-baseline, indoor pose estimation on ScanNet. Through end-to-end training with pose optimization, our network learns to predict valuable matches for pose estimation, and downweights outliers. This enables accurate weighted pose estimation, which outperforms the baselines. “cross-dataset” indicates that COTR was trained on MegaDepth.

compute the area under the curve (AUC) in % at the thresholds  $[5^\circ, 10^\circ, 20^\circ]$  of the pose error, i.e., the maximum of rotation and translation error, where the translation error is the angle between translation vectors, since poses are only determined up to an unknown scale factor. Tabs. 1 and 2 list the AUC metrics for four pose estimation methods: (i) essential matrix estimation with RANSAC, (ii) weighted eight-point algorithm (Sec. 3.2), (iii) RANSAC followed by  $T = 10$  bundle adjustment iterations (Sec. 3.2) and (iv) weighted eight-point algorithm followed by  $T = 10$  bundle adjustment iterations (Sec. 3.2). The results show that our method outperforms the baselines on two-view pose estimation. For our method, the combination of weighted eight-point algorithm and bundle adjustment is stronger than pose estimation with RANSAC in the indoor and outdoor setting. This shows that end-to-end training enables the learning of accurate confidences that down-weight outliers and render RANSAC unnecessary.

	Pose est. method	Pose error AUC [%] $\uparrow$		
		@5°	@10°	@20°
Mutual nearest neighbor	RANSAC	32.2	47.6	55.2
SuperGlue [45]		43.4	61.6	76.2
LoFTR [48]		52.8	69.2	81.2
COTR [26]		35.2	53.9	69.6
3DG-STFM [35]		52.6	68.5	80.0
Ours w/o multi-view		49.5	66.7	79.9
Mutual nearest neighbor	Weight. 8-point	0.1	0.2	1.0
SuperGlue [45]		23.8	36.2	49.2
LoFTR [48]		15.5	27.1	41.6
COTR [26]		29.6	43.4	57.2
3DG-STFM [35]		4.0	9.5	19.8
Ours w/o multi-view		46.9	62.8	76.3
Mutual nearest neighbor	RANSAC + bundle adjust.	34.9	49.5	61.9
SuperGlue [45]		48.3	65.2	78.3
LoFTR [48]		52.8	69.6	82.0
COTR [26] cross-dataset		45.0	61.1	73.8
3DG-STFM [35]		51.2	67.7	80.2
Ours w/o multi-view		55.3	70.8	82.3
Mutual nearest neighbor	Weight. 8-point + bundle adjust.	0.1	0.8	4.3
SuperGlue [45]		40.3	53.6	65.6
LoFTR [48]		25.7	40.0	54.7
COTR [26]		47.1	61.3	72.5
3DG-STFM [35]		10.2	20.0	35.0
Ours w/o multi-view		<b>61.2</b>	<b>74.9</b>	<b>85.0</b>

Table 2. Baseline comparison on two-view, wide-baseline, outdoor pose estimation on MegaDepth. The pose optimization objective guides our method to produce matches with accurate confidences for weighted pose estimation, leading to higher pose accuracy than the baselines relying on RANSAC.

## 4.2. Multi-View Pose Estimation

For multi-view evaluation, we sample test images with the same overlap criterion as used by prior work to sample image pairs [45, 48, 35]. However, instead of sampling a pair, we sample a 5-tuple, by appending three more images that each satisfy the overlap criterion to the previous one. Further detail and overlap ranges are provided in the supplement. Besides ScanNet and MegaDepth, we evaluate on Matterport3D, which is particularly challenging for matching, as view captures are much more sparse, i.e., neighboring images are  $60^\circ$  horizontally and  $30^\circ$  vertically apart. This difficult dataset, serves to measure robustness on the pose estimation task.

Multi-view pose estimation is evaluated as follows: (i) Feature matches are computed. Baselines that operate on image pairs are run on all possible pairs of the tuple. (ii) Relative poses are estimated between all possible pairs using the best performing two-view pose estimation from Sec. 4.1. (iii) Absolute poses are determined through robust estimators for rotation [11] and translation [37], which take

	Transl. error AUC [%] ↑			Rot. error AUC [%] ↑		
	@5°	@10°	@20°	@5°	@10°	@20°
Mutual nearest neighbor	8.5	17.8	31.0	33.0	48.4	62.8
SuperGlue [45]	21.3	37.5	53.7	54.2	71.0	82.6
LoFTR [48]	20.6	36.9	53.7	57.3	72.0	82.0
COTR [26] cross-dataset	10.9	22.4	36.9	38.8	53.6	66.3
3DG-STFM [35]	22.0	38.7	55.5	57.0	72.7	83.0
Ours	<b>26.9</b>	<b>45.6</b>	<b>63.0</b>	<b>64.2</b>	<b>78.8</b>	<b>87.7</b>

Table 3. Baseline comparison on multi-view indoor pose estimation on ScanNet. Our multi-view and end-to-end approach, predicts matches and confidences that improve pose estimation compared to the pairwise baselines. “cross-dataset” indicates that COTR was trained on MegaDepth.

	Transl. error AUC [%] ↑			Rot. error AUC [%] ↑		
	@5°	@10°	@20°	@5°	@10°	@20°
Mutual nearest neighbor	2.8	5.6	10.6	3.3	6.6	12.3
SuperGlue [45]	17.1	24.0	32.7	17.9	25.9	35.3
Ours w/o multi-view	19.4	27.8	38.4	20.9	30.5	41.8
Ours w/o end-to-end	28.5	35.4	42.7	29.4	38.0	46.2
Ours	<b>33.2</b>	<b>42.1</b>	<b>51.6</b>	<b>35.1</b>	<b>45.8</b>	<b>56.2</b>

Table 4. Baseline comparison and ablation study on multi-view indoor pose estimation on Matterport3D. The full version of our method, with multi-view matching and end-to-end training with pose optimization, achieves best performance.

	Transl. error AUC [%] ↑			Rot. error AUC [%] ↑		
	@5°	@10°	@20°	@5°	@10°	@20°
Mutual nearest neighbor	12.0	20.1	31.9	23.4	36.7	51.8
SuperGlue [45]	47.3	58.7	68.9	60.9	73.6	83.4
LoFTR [48]	48.7	59.5	69.5	63.9	75.3	84.0
COTR [26]	37.9	48.1	58.3	49.8	61.9	72.7
3DG-STFM [35]	44.5	55.3	65.8	59.5	71.9	81.7
Ours	<b>52.1</b>	<b>63.0</b>	<b>72.5</b>	<b>66.7</b>	<b>77.8</b>	<b>85.9</b>

Table 5. Baseline comparison on multi-view outdoor pose estimation on MegaDepth. Through multi-view matching and end-to-end training, our method achieves higher pose estimation accuracy than the baselines.

initial absolute poses and relative poses as input. The initial absolute poses are obtained by composing relative poses along edges of a maximum spanning tree on the match graph, where edge weights are inlier counts from the previous step. (iv) Bundle adjustment jointly optimizes all poses by minimizing the confidence-weighted reprojection error of inlier matches using Ceres Solver for non-linear least squares optimization [2]. The pose estimation performance is measured by the translation and rotation error AUC between all possible pairs of the tuple.

The quantitative results (Tabs. 3 to 5) show that our method achieves higher AUC metrics than the baselines across all thresholds in the indoor and outdoor setting. The metrics on Matterport3D are overall lower than on ScanNet and MegaDepth, due to the smaller overlap between images.

	Pose error AUC [%] ↑	
	@5°	@10°
SuperGlue [45]	70.0	80.2
Ours	<b>74.5</b>	<b>83.4</b>

Table 6. IMC multi-view evaluation using COLMAP SfM on the PhotoTourism dataset. Although COLMAP does not use matching confidences, there is a clear benefit from our multi-view matching method.

In this scenario, our method outperforms SuperGlue with a larger gap than on ScanNet or MegaDepth, which shows that our approach copes better with the more challenging setting in Matterport3D. For qualitative comparison, we visualize the reprojection error by projecting the ground truth depth maps from all other views using the estimated poses, scaled according to the ground truth (Figs. 3 and 4). With multi-view reasoning during matching and learned outlier rejection through end-to-end training, our method is robust to challenging situations, like repetitive patterns (Fig. 3 sample 2) or large viewpoint changes (Fig. 3 sample 1).

We further evaluate multi-view pose estimation using the protocol of the Image Matching Challenge (IMC) 2021 [27]. It provides a multi-view setting, where COLMAP [46] Structure-from-Motion (SfM) estimates camera poses on groups of 5-25 internet images of tourist attractions. Tab. 6 lists the pose error AUC metrics for the detector-based methods, SuperGlue and Ours. Even though COLMAP does not consider our learned confidence weights, we observe a clear improvement through our end-to-end and multi-view approach.

Details on the baseline comparisons, further qualitative results and a cross-attention visualization are provided in the supplementary material.

### 4.3. Runtime

Tab. 7 compares runtime for matching and pose estimation. Our method requires the same amount of time as SuperGlue for matching an image pair, however, we reduce runtime by 9% when matching a 5-tuple. The savings stem from fewer intra-frame GNN messages in multi-view matching compared to matching the corresponding pairs individually (see supplementary material). The detector-free baselines take far more time for matching. Our method more than halves the RANSAC time compared to SuperGlue. This shows that our confidences allow for better outlier pre-filtering by confidence thresholding, which improves the ratio between inliers and outliers prior to RANSAC. Our proposed weighted pose estimation (weighted eight-point + bundle adjustment)—besides reducing the pose error (Sec. 4.1)—reduces the runtime on SuperGlue matches and our matches by half, compared to RANSAC on SuperGlue matches. Only COTR, due to a



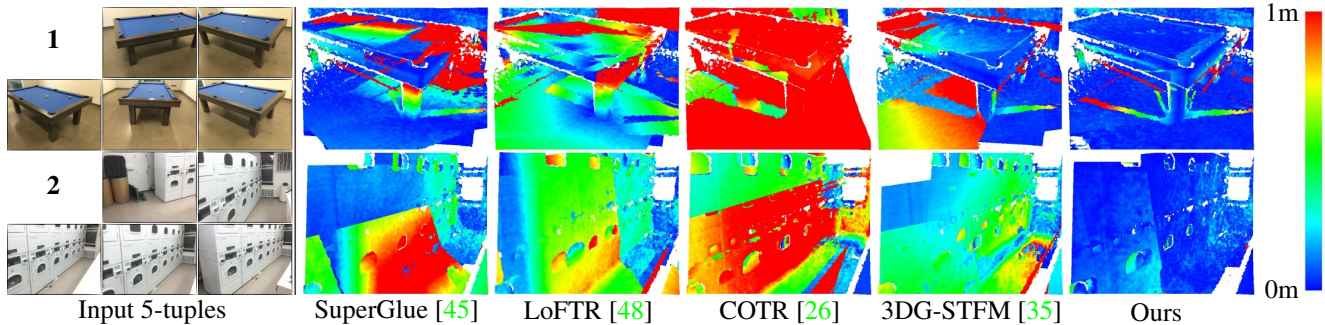


Figure 3. Reprojection error (right) for estimated camera poses on ScanNet 5-tuples (left). With multi-view matching and end-to-end training, our method successfully handles challenging pose estimation scenarios, while baselines have severe camera pose errors.

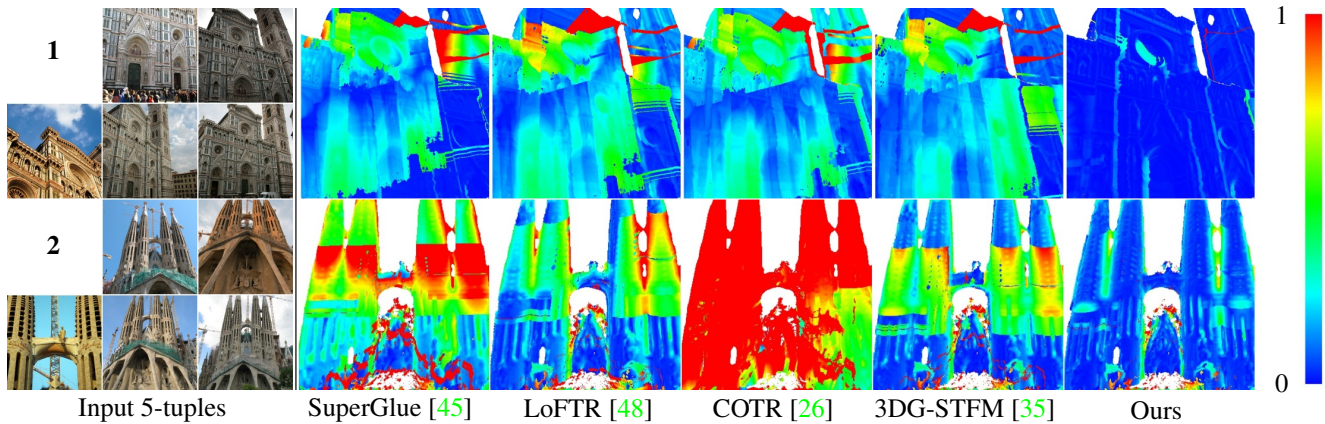


Figure 4. Reprojection error (right) for estimated camera poses on MegaDepth 5-tuples (left). Through multi-view matching and end-to-end training, our method successfully estimates camera poses in challenging outdoor scenarios, while baselines show misalignment. Reprojection errors are visualized in the MegaDepth scaling.

smaller number of matches, has a shorter pose estimation runtime, however, its matching time is multiple orders of magnitude higher and the pose accuracy is lower. All runtime is measured on a Nvidia GeForce RTX 2080. For a fair comparison to the detector-free matchers, the matching time of SuperGlue and our method includes the SuperPoint inference time.

	Matching time ↓		Pose estimation time ↓		
	2-view	5-view ≥ 10 pairs	RANSAC	Weight. 8-point	Bundle adjust.
SuperGlue [45]	60 ms	371 ms	126 ms	5 ms	56 ms
LoFTR [48]	108 ms	976 ms	148 ms	9 ms	511 ms
COTR [26]	37950 ms	357096 ms	126 ms	5 ms	47 ms
3DG-STFM [35]	130 ms	1176 ms	201 ms	10 ms	735 ms
Ours	60 ms	338 ms	52 ms	5 ms	56 ms

Table 7. Matching and pose estimation time on ScanNet. Multi-view matching is faster than matching the corresponding pairs. Our confidences enable effective thresholding prior to RANSAC, reducing its runtime. Weighted eight-point + bundle adjustment is faster or comparable to RANSAC on SuperGlue and our matches.

#### 4.4. Ablation Study

The quantitative results on Matterport3D (Tab. 4) show that the full version of our method achieves the best performance. This is consistent with the qualitative results (Fig. 5), as well as the ablation results on ScanNet and MegaDepth, which are provided in the supplement.

**Without Multi-View.** Omitting multi-view in the GNN causes an average performance drop of 14.2% on Matterport3D. This suggests that the multi-view receptive field supports information flow from other views to bridge gaps, where the overlap is small. Sample 1 in Fig. 5 shows that without multi-view reasoning, the matching fails to resolve large viewpoint changes and difficult object symmetries.

**Without End-to-End.** Without end-to-end training the average performance drops by 7.3%. This shows that end-to-end training enables the learning of an outlier down-weighting, that improves pose estimation. Dropping end-to-end leads to increased misalignment in Fig. 5.



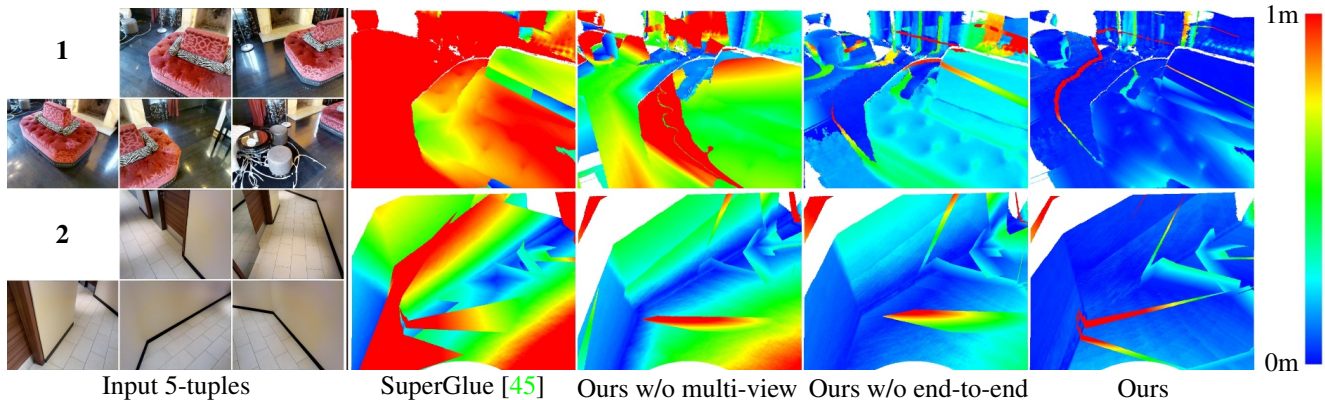


Figure 5. Reprojection error (right) for estimated camera poses on Matterport3D 5-tuples (left). Our complete method improves camera alignment over the ablated versions and SuperGlue, showing the importance of multi-view matching and end-to-end training.

**Variable Number of Input Views.** In Fig. 6, we investigate the impact of the number of images used for matching, both in pairwise (w/o multi-view) and joint (w/ multi-view) manner. The experiment is conducted on sequences of 9 images which are generated on ScanNet as described in Sec. 4.2. The results show that pose estimation improves, when matching across a larger span of neighboring images. The curves, however, plateau when a larger window size does not bring any more relevant images into the matching. Additionally, the results show the benefit of joint matching in a single graph as opposed to matching all possible image pairs individually.

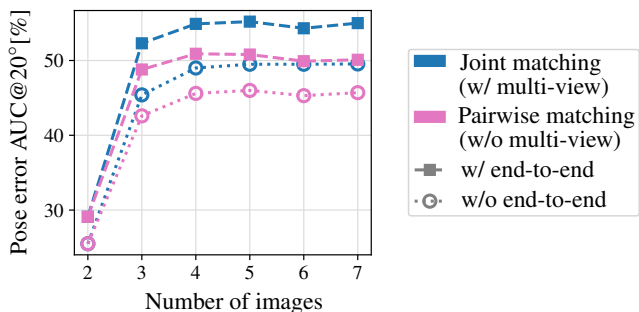


Figure 6. Pose error AUC on sequences of 9 images on ScanNet using variable number of images in pairwise or joint matching. Multi-view matching across  $\sim 5$  images combined with end-to-end training gives the best performance.

**Variable Image Overlap.** Evaluations on reduced image overlap are provided in the supplementary material.

#### 4.5. Limitations

One of our contributions is the end-to-end differentiability of the pose optimization that guides the matching network. While this significantly improves the pose estimation

results, we currently only backpropagate gradients to the matching network, but do not update keypoint descriptors; i.e., we use existing SuperPoint [16]. However, we believe that jointly training feature descriptors is a promising avenue to even further improve performance. Besides, more recent keypoint detectors and descriptors like ASLFeat [33], in contrast to SuperPoint, provide subpixel accuracy, which can boost subsequent matching and pose estimation.

## 5. Conclusion

We have presented a method that couples multi-view feature matching and pose optimization into an end-to-end trainable pipeline. Our graph neural network matches features across multiple views in a joint fashion, which enables globally informed matching solutions. Combined with differentiable pose optimization, gradients inform the matching network, which learns to reject outliers to produce valuable matches for pose estimation. Our method significantly improves pose estimation compared to prior work. In particular, we observe increased robustness in challenging settings, such as in presence of repetitive structure or small image overlap as in the Matterport3D dataset. Overall, we believe that our end-to-end approach is an important stepping stone towards an end-to-end trained SLAM method.

## Acknowledgements

This work was supported by the ERC Starting Grant Scan2CAD (804724), the German Research Foundation (DFG) Grant “Making Machine Learning on Static and Dynamic 3D Data Practical”, and the German Research Foundation (DFG) Research Unit “Learning and Simulation in Visual Computing”. We thank Angela Dai for the video voice over.

## References

- [1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 2
- [2] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 3 2022. 7
- [3] Dániel Baráth and Jiri Matas. Magsac: Marginalizing sample consensus. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10189–10197, 2019. 2
- [4] Aritra Bhowmik, Stefan Gumhold, Carsten Rother, and Eric Brachmann. Reinforced feature points: Optimizing feature detection and description for a high-level task. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4947–4956, 2020. 2
- [5] 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. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2828–2837, 2017. 2
- [6] Jose Luis Blanco. A tutorial on se(3) transformation parameterizations and on-manifold optimization. *University of Malaga, Tech. Rep.*, 09 2010. 5
- [7] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4321–4330, 2019. 2
- [8] Luca Cavalli, Viktor Larsson, Martin R. Oswald, Torsten Sattler, and Marc Pollefeys. Handcrafted outlier detection revisited. In *ECCV*, 2020. 2
- [9] Jan Cech, Jiri Matas, and Michal Perdoch. Efficient sequential correspondence selection by cosegmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1568–1581, 2008. 2
- [10] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *3DV*, 2017. 2
- [11] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *2013 IEEE International Conference on Computer Vision*, pages 521–528, 2013. 6
- [12] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2
- [13] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, 2013. 4
- [14] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. *CVPR*, 2017. 2
- [15] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface face reintegration. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017. 2
- [16] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 337–33712, 2018. 2, 3, 5, 9
- [17] Mihai Dusmanu, Ignacio Rocco, Tomáš Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8084–8093, 2019. 2
- [18] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in Neural Information Processing Systems (NIPS)*, 2015. 3
- [19] 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*, 24:381–395, 1981. 2
- [20] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. Convolutional sequence to sequence learning. In *ICML*, 2017. 3
- [21] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017. 3
- [22] Lei Han, Mengqi Ji, Lu Fang, and Matthias Nießner. RegNet: Learning the optimization of direct image-to-image pose registration. *arXiv preprint arXiv:1812.10212*, 2018. 2
- [23] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997. 4
- [24] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004. 4
- [25] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2965–2973, 2015. 4
- [26] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. COTR: Correspondence Transformer for Matching Across Images. In *ICCV*, 2021. 1, 2, 5, 6, 7, 8
- [27] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129(2):517–547, 2021. 7
- [28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 2015. 5
- [29] Xinghui Li, K. Han, Shuda Li, and Victor Adrian Prisacariu. Dual-resolution correspondence networks. *NeurIPS*, 2020. 2
- [30] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 2

- [31] Hugh Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981. [4](#)
- [32] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004. [2](#)
- [33] Zixin Luo, Lei Zhou, Xuyang Bai, Hongkai Chen, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Aslfeat: Learning local features of accurate shape and localization. *Computer Vision and Pattern Recognition (CVPR)*, 2020. [9](#)
- [34] Jiayi Ma, Ji Zhao, Junjun Jiang, Huabing Zhou, and Xiaojie Guo. Locality preserving matching. *International Journal of Computer Vision*, pages 512–531, 2018. [2](#)
- [35] Runyu Mao, Chen Bai, Yatong An, Fengqing Zhu, and Cheng Lu. 3dg-stfm: 3d geometric guided student-teacher feature matching. *ECCV*, 2022. [1](#), [2](#), [5](#), [6](#), [7](#), [8](#)
- [36] Yuki Ono, Eduard Trulls, Pascal V. Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. In *NeurIPS*, 2018. [2](#)
- [37] Onur Özyesil and Amit Singer. Robust camera location estimation by convex programming. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2674–2683, 2015. [6](#)
- [38] G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, Rama Chellappa, and Pedro Miraldo. 3dregnet: A deep neural network for 3d point registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7193–7203, 2020. [2](#)
- [39] Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In *ECCV*, 2008. [2](#)
- [40] René Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *ECCV*, 2018. [2](#)
- [41] Jérôme Revaud, Philippe Weinzaepfel, César Roberto de Souza, Noé Pion, Gabriela Csurka, Johann Cabon, and M. Humenberger. R2d2: Repeatable and reliable detector and descriptor. *Advances in Neural Information Processing Systems*, 2019. [2](#)
- [42] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *ECCV*, 2020. [2](#)
- [43] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomás Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *NeurIPS*, 2018. [2](#)
- [44] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski. Orb: An efficient alternative to sift or surf. *2011 International Conference on Computer Vision*, pages 2564–2571, 2011. [2](#)
- [45] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4937–4946, 2020. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#)
- [46] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [7](#)
- [47] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, pages 343–348, 1967. [4](#)
- [48] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8918–8927, 2021. [1](#), [2](#), [5](#), [6](#), [7](#), [8](#)
- [49] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *arXiv preprint arXiv:1806.04807*, 2018. [2](#)
- [50] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999. [2](#)
- [51] Tinne Tuytelaars and Luc Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *BMVC*, 2000. [2](#)
- [52] Michal J. Tyszkiewicz, P. Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 2020. [2](#)
- [53] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017. [2](#)
- [54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. [2](#), [3](#)
- [55] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal V. Fua. Lift: Learned invariant feature transform. *ECCV*, 2016. [2](#)
- [56] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal V. Fua. Learning to find good correspondences. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2666–2674, 2018. [2](#)
- [57] Christopher Zach. Robust bundle adjustment revisited. In *European Conference on Computer Vision*, pages 772–787. Springer, 2014. [2](#)
- [58] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Honggen Liao. Learning two-view correspondences and geometry using order-aware network. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5844–5853, 2019. [2](#)