SuperGlue: Learning Feature Matching with Graph Neural Networks

Paul-Edouard Sarlin[1*] Daniel DeTone[2] Tomasz Malisiewicz[2] Andrew Rabinovich[2]
[1] ETH Zurich [2] Magic Leap, Inc.

Abstract

This paper introduces SuperGlue, a neural network that matches two sets of local features by jointly finding correspondences and rejecting non-matchable points. Assignments are estimated by solving a differentiable optimal transport problem, whose costs are predicted by a graph neural network. We introduce a flexible context aggregation mechanism based on attention, enabling SuperGlue to reason about the underlying 3D scene and feature assignments jointly. Compared to traditional, hand-designed heuristics, our technique learns priors over geometric transformations and regularities of the 3D world through end-to-end training from image pairs. SuperGlue outperforms other learned approaches and achieves state-of-the-art results on the task of pose estimation in challenging real-world indoor and outdoor environments. The proposed method performs matching in real-time on a modern GPU and can be readily integrated into modern SfM or SLAM systems. The code and trained weights are publicly available at `github.com/magicleap/SuperGluePretrainedNetwork`.

1. Introduction

Correspondences between points in images are essential for estimating the 3D structure and camera poses in geometric computer vision tasks such as Simultaneous Localization and Mapping (SLAM) and Structure-from-Motion (SfM). Such correspondences are generally estimated by matching local features, a process known as data association. Large viewpoint and lighting changes, occlusion, blur, and lack of texture are factors that make 2D-to-2D data association particularly challenging.

In this paper, we present a new way of thinking about the feature matching problem. Instead of learning better task-agnostic local features followed by simple matching heuristics and tricks, we propose to learn the matching process from pre-existing local features using a novel neural architecture called SuperGlue. In the context of SLAM, which typically [8] decomposes the problem into the visual feature extraction *front-end* and the bundle adjustment or pose estimation *back-end*, our network lies directly in the middle – SuperGlue is a learnable *middle-end* (see Figure 1).

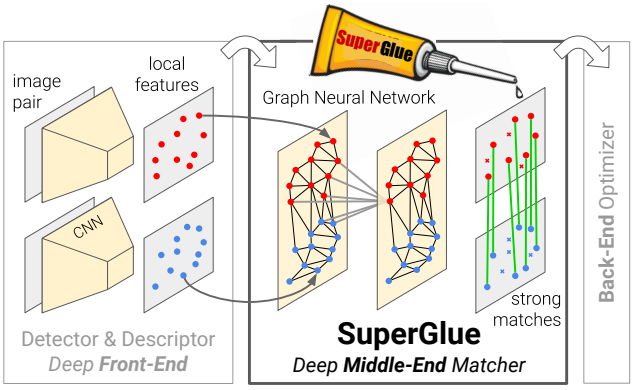

Figure 1: **Feature matching with SuperGlue.** Our approach establishes pointwise correspondences from off-the-shelf local features: it acts as a middle-end between hand-crafted or learned front-end and back-end. SuperGlue uses a graph neural network and attention to solve an assignment optimization problem, and handles partial point visibility and occlusion elegantly, producing a partial assignment.

In this work, *learning feature matching* is viewed as finding the partial assignment between two sets of local features. We revisit the classical graph-based strategy of matching by solving a linear assignment problem, which, when relaxed to an optimal transport problem, can be solved differentiably. The cost function of this optimization is predicted by a Graph Neural Network (GNN). Inspired by the success of the Transformer [61], it uses self- (intra-image) and cross- (inter-image) attention to leverage both spatial relationships of the keypoints and their visual appearance. This formulation enforces the assignment structure of the predictions while enabling the cost to learn complex priors, elegantly handling occlusion and non-repeatable keypoints. Our method is trained end-to-end from image pairs – we learn priors for pose estimation from a large annotated dataset, enabling SuperGlue to reason about the 3D scene and the assignment. Our work can be applied to a variety of multiple-view geometry problems that require high-quality feature correspondences (see Figure 2).

*Work done at Magic Leap, Inc. for a Master's degree. The author thanks his academic supervisors: Cesar Cadena, Marcin Dymczyk, Juan Nieto.

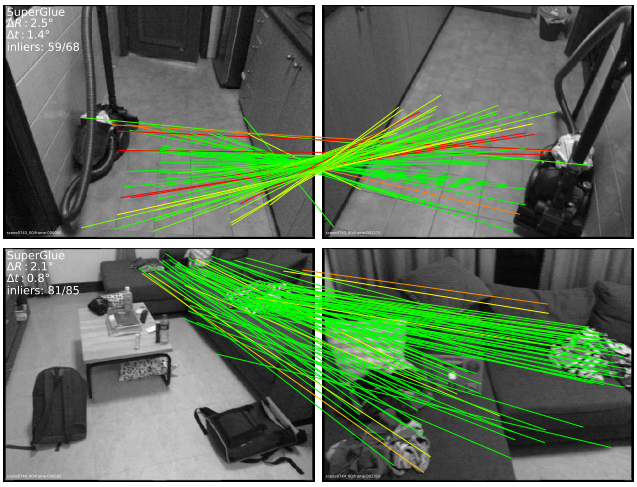

Figure 2: **SuperGlue correspondences.** For these two challenging indoor image pairs, matching with SuperGlue results in accurate poses while other learned or handcrafted methods fail (correspondences colored by epipolar error).

We show the superiority of SuperGlue compared to both handcrafted matchers and learned inlier classifiers. When combined with SuperPoint [18], a deep front-end, SuperGlue advances the state-of-the-art on the tasks of indoor and outdoor pose estimation and paves the way towards end-to-end deep SLAM.

2. Related work

Local feature matching is generally performed by i) detecting interest points, ii) computing visual descriptors, iii) matching these with a Nearest Neighbor (NN) search, iv) filtering incorrect matches, and finally v) estimating a geometric transformation. The classical pipeline developed in the 2000s is often based on SIFT [31], filters matches with Lowe's ratio test [31], the mutual check, and heuristics such as neighborhood consensus [59, 10, 6, 49], and finds a transformation with a robust solver like RANSAC [21, 43].

Recent works on deep learning for matching often focus on learning better sparse detectors and local descriptors [18, 19, 37, 45, 69] from data using Convolutional Neural Networks (CNNs). To improve their discriminativeness, some works explicitly look at a wider context using regional features [32] or log-polar patches [20]. Other approaches learn to filter matches by classifying them into inliers and outliers [33, 44, 7, 71]. These operate on sets of matches, still estimated by NN search, and thus ignore the assignment structure and discard visual information. Works that learn to perform matching have so far focused on dense matching [46] or 3D point clouds [65], and still exhibit the same limitations. In contrast, our learnable middle-end simultaneously performs context aggregation, matching, and filtering in a single end-to-end architecture.

Graph matching problems are usually formulated as quadratic assignment problems, which are NP-hard, requiring expensive, complex, and thus impractical solvers [30]. For local features, the computer vision literature of the 2000s [5, 27, 57] uses handcrafted costs with many heuristics, making it complex and brittle. Caetano *et al*. [9] learn the cost of the optimization for a simpler linear assignment, but only use a shallow model, while our SuperGlue learns a flexible cost using a deep neural network. Related to graph matching is the problem of *optimal transport* [63] – it is a generalized linear assignment with an efficient yet simple approximate solution, the Sinkhorn algorithm [55, 12, 39].

Deep learning for sets such as point clouds aims at designing permutation equi- or invariant functions by aggregating information across elements. Some works treat all elements equally, through global pooling [70, 40, 15] or instance normalization [60, 33, 32], while others focus on a local neighborhood in coordinate or feature space [41, 66]. Attention [61, 64, 62, 26] can perform both global and data-dependent local aggregation by focusing on specific elements and attributes, and is thus more flexible. By observing that self-attention can be seen as an instance of a Message Passing Graph Neural Network [23, 4] on a complete graph, we apply attention to graphs with multiple types of edges, similar to [28, 72], and enable SuperGlue to learn complex reasoning about the two sets of local features.

3. The SuperGlue Architecture

Motivation: In the image matching problem, some regularities of the world could be leveraged: the 3D world is largely smooth and sometimes planar, all correspondences for a given image pair derive from a single epipolar transform if the scene is static, and some poses are more likely than others. In addition, 2D keypoints are usually projections of salient 3D points, like corners or blobs, thus correspondences across images must adhere to certain physical constraints: i) a keypoint can have at most a single correspondence in the other image; and ii) some keypoints will be unmatched due to occlusion and failure of the detector. An effective model for feature matching should aim at finding all correspondences between reprojections of the same 3D points and identifying keypoints that have no matches. We formulate SuperGlue (see Figure 3) as solving an optimization problem, whose cost is predicted by a deep neural network. This alleviates the need for domain expertise and heuristics – we learn relevant priors directly from the data.

Formulation: Consider two images A and B, each with a set of keypoint *positions* $\mathbf{p}$ and associated *visual descriptors* $\mathbf{d}$ – we refer to them jointly $(\mathbf{p}, \mathbf{d})$ as the *local features*. Positions consist of x and y image coordinates as well as a detection confidence c, $\mathbf{p}_i := (x, y, c)_i$. Visual descriptors $\mathbf{d}_i \in \mathbb{R}^D$ can be those extracted by a CNN like SuperPoint

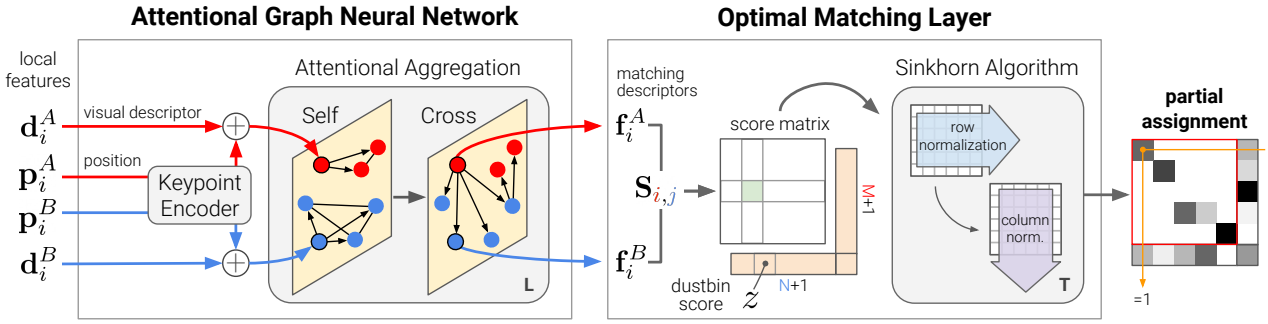

Figure 3: **The SuperGlue architecture.** SuperGlue is made up of two major components: the *attentional graph neural network* (Section 3.1), and the *optimal matching layer* (Section 3.2). The first component uses a *keypoint encoder* to map keypoint positions $\mathbf{p}$ and their visual descriptors $\mathbf{d}$ into a single vector, and then uses alternating self- and cross-attention layers (repeated L times) to create more powerful representations $\mathbf{f}$. The optimal matching layer creates an M by N score matrix, augments it with dustbins, then finds the optimal partial assignment using the Sinkhorn algorithm (for T iterations).

or traditional descriptors like SIFT. Images A and B have M and N local features, indexed by $\mathcal{A} := \{1, ..., M\}$ and $\mathcal{B} := \{1, ..., N\}$, respectively.

Partial Assignment: Constraints i) and ii) mean that correspondences derive from a partial assignment between the two sets of keypoints. For the integration into downstream tasks and better interpretability, each possible correspondence should have a confidence value. We consequently define a partial soft assignment matrix $\mathbf{P} \in [0,1]^{M \times N}$ as:

$$\mathbf{P}\mathbf{1}_N \leq \mathbf{1}_M \quad \text{and} \quad \mathbf{P}^\top \mathbf{1}_M \leq \mathbf{1}_N. \tag{1}$$

Our goal is to design a neural network that predicts the assignment $\mathbf{P}$ from two sets of local features.

3.1. Attentional Graph Neural Network

Besides the position of a keypoint and its visual appearance, integrating other contextual cues can intuitively increase its distinctiveness. We can for example consider its spatial and visual relationship with other co-visible keypoints, such as ones that are salient [32], self-similar [54], statistically co-occurring [73], or adjacent [58]. On the other hand, knowledge of keypoints in the second image can help to resolve ambiguities by comparing candidate matches or estimating the relative photometric or geometric transformation from global and unambiguous cues.

When asked to match a given ambiguous keypoint, humans look back-and-forth at both images: they sift through tentative matching keypoints, examine each, and look for contextual cues that help disambiguate the true match from other self-similarities [11]. This hints at an iterative process that can focus its attention on specific locations.

We consequently design the first major block of SuperGlue as an Attentional Graph Neural Network (see Figure 3). Given initial local features, it computes *matching descriptors* $\mathbf{f}_i \in \mathbb{R}^D$ by letting the features communicate with each other. As we will show, long-range feature aggregation within and across images is vital for robust matching.

Keypoint Encoder: The initial representation ${}^{(0)}\mathbf{x}_i$ for each keypoint i combines its visual appearance and location. We embed the keypoint position into a high-dimensional vector with a Multilayer Perceptron (MLP) as:

$${}^{(0)}\mathbf{x}_i = \mathbf{d}_i + \text{MLP}_{\text{enc}}\left(\mathbf{p}_i\right). \tag{2}$$

This encoder enables the graph network to later reason about both appearance and position jointly, especially when combined with attention, and is an instance of the "positional encoder" popular in language processing [22, 61].

Multiplex Graph Neural Network: We consider a single complete graph whose nodes are the keypoints of both images. The graph has two types of undirected edges – it is a *multiplex graph* [34, 36]. Intra-image edges, or *self* edges, $\mathcal{E}_{\text{self}}$, connect keypoints i to all other keypoints within the same image. Inter-image edges, or *cross* edges, $\mathcal{E}_{\text{cross}}$, connect keypoints i to all keypoints in the other image. We use the message passing formulation [23, 4] to propagate information along both types of edges. The resulting multiplex Graph Neural Network starts with a high-dimensional state for each node and computes at each layer an updated representation by simultaneously aggregating messages across all given edges for all nodes.

Let ${}^{(\ell)}\mathbf{x}_i^A$ be the intermediate representation for element i in image A at layer ℓ. The message $\mathbf{m}_{\mathcal{E}\to i}$ is the result of the aggregation from all keypoints $\{j : (i,j) \in \mathcal{E}\}$, where $\mathcal{E} \in \{\mathcal{E}_{\text{self}}, \mathcal{E}_{\text{cross}}\}$. The residual message passing update for all i in A is:

$${}^{(\ell+1)}\mathbf{x}_i^A = {}^{(\ell)}\mathbf{x}_i^A + \text{MLP}\left(\left[{}^{(\ell)}\mathbf{x}_i^A \,||\, \mathbf{m}_{\mathcal{E}\to i}\right]\right), \tag{3}$$

where $[\cdot \,||\, \cdot]$ denotes concatenation. A similar update can be simultaneously performed for all keypoints in image B. A fixed number of layers L with different parameters are chained and alternatively aggregate along the self and cross edges. As such, starting from $\ell = 1$, $\mathcal{E} = \mathcal{E}_{\text{self}}$ if ℓ is odd and $\mathcal{E} = \mathcal{E}_{\text{cross}}$ if ℓ is even.

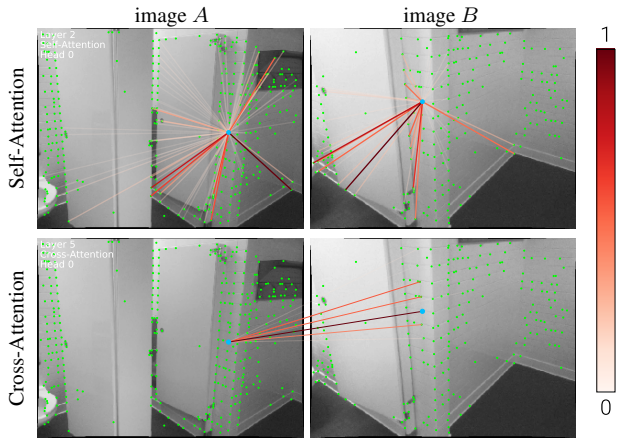

Figure 4: **Visualizing self- and cross-attention.** Attentional aggregation builds a dynamic graph between keypoints. Weights α_{ij} are shown as rays. Self-attention (top) can attend anywhere in the same image, e.g. distinctive locations, and is thus not restricted to nearby locations. Cross-attention (bottom) attends to locations in the other image, such as potential matches that have a similar appearance.

Attentional Aggregation: An attention mechanism performs the aggregation and computes the message $\mathbf{m}_{\mathcal{E}\to i}$. Self edges are based on self-attention [61] and cross edges are based on *cross-attention*. Akin to database retrieval, a representation of i, the query $\mathbf{q}_i$, retrieves the values $\mathbf{v}_j$ of some elements based on their attributes, the keys $\mathbf{k}_j$. The message is computed as a weighted average of the values:

$$\mathbf{m}_{\mathcal{E}\to i} = \sum_{j:(i,j)\in\mathcal{E}} \alpha_{ij}\mathbf{v}_j, \tag{4}$$

where the attention weight α_{ij} is the Softmax over the key-query similarities: $\alpha_{ij} = \text{Softmax}_j\left(\mathbf{q}_i^\top \mathbf{k}_j\right)$.

The key, query, and value are computed as linear projections of deep features of the graph neural network. Considering that query keypoint i is in the image Q and all source keypoints are in image S, $(Q,S)\in\{A,B\}^2$, we can write:

$$\begin{aligned}\mathbf{q}_i &= \mathbf{W}_1\, {}^{(\ell)}\mathbf{x}_i^Q + \mathbf{b}_1\\ \begin{bmatrix}\mathbf{k}_j\\ \mathbf{v}_j\end{bmatrix} &= \begin{bmatrix}\mathbf{W}_2\\ \mathbf{W}_3\end{bmatrix} {}^{(\ell)}\mathbf{x}_j^S + \begin{bmatrix}\mathbf{b}_2\\ \mathbf{b}_3\end{bmatrix}.\end{aligned} \tag{5}$$

Each layer ℓ has its own projection parameters, learned and shared for all keypoints of both images. In practice, we improve the expressivity with multi-head attention [61].

Our formulation provides maximum flexibility as the network can learn to focus on a subset of keypoints based on specific attributes (see Figure 4). SuperGlue can retrieve or attend based on both appearance and keypoint location as they are encoded in the representation $\mathbf{x}_i$. This includes attending to a nearby keypoint and retrieving the relative positions of similar or salient keypoints. This enables representations of the geometric transformation and the assignment. The final matching descriptors are linear projections:

$$\mathbf{f}_i^A = \mathbf{W}\cdot {}^{(L)}\mathbf{x}_i^A + \mathbf{b}, \quad \forall i\in\mathcal{A}, \tag{6}$$

and similarly for keypoints in B.

3.2. Optimal matching layer

The second major block of SuperGlue (see Figure 3) is the optimal matching layer, which produces a partial assignment matrix. As in the standard graph matching formulation, the assignment $\mathbf{P}$ can be obtained by computing a score matrix $\mathbf{S}\in\mathbb{R}^{M\times N}$ for all possible matches and maximizing the total score $\sum_{i,j}\mathbf{S}_{i,j}\mathbf{P}_{i,j}$ under the constraints in Equation 1. This is equivalent to solving a linear assignment problem.

Score Prediction: Building a separate representation for all $M\times N$ potential matches would be prohibitive. We instead express the pairwise score as the similarity of matching descriptors:

$$\mathbf{S}_{i,j} = < \mathbf{f}_i^A, \mathbf{f}_j^B >, \ \forall (i,j)\in\mathcal{A}\times\mathcal{B}, \tag{7}$$

where $<\cdot,\cdot>$ is the inner product. As opposed to learned visual descriptors, the matching descriptors are not normalized, and their magnitude can change per feature and during training to reflect the prediction confidence.

Occlusion and Visibility: To let the network suppress some keypoints, we augment each set with a dustbin so that unmatched keypoints are explicitly assigned to it. This technique is common in graph matching, and dustbins have also been used by SuperPoint [18] to account for image cells that might not have a detection. We augment the scores $\mathbf{S}$ to $\bar{\mathbf{S}}$ by appending a new row and column, the point-to-bin and bin-to-bin scores, filled with a single learnable parameter:

$$\bar{\mathbf{S}}_{i,N+1} = \bar{\mathbf{S}}_{M+1,j} = \bar{\mathbf{S}}_{M+1,N+1} = z\in\mathbb{R}. \tag{8}$$

While keypoints in A will be assigned to a single keypoint in B or the dustbin, each dustbin has as many matches as there are keypoints in the other set: N, M for dustbins in A, B respectively. We denote as $\mathbf{a} = \begin{bmatrix}\mathbf{1}_M^\top & N\end{bmatrix}^\top$ and $\mathbf{b} = \begin{bmatrix}\mathbf{1}_N^\top & M\end{bmatrix}^\top$ the number of expected matches for each keypoint and dustbin in A and B. The augmented assignment $\bar{\mathbf{P}}$ now has the constraints:

$$\bar{\mathbf{P}}\mathbf{1}_{N+1} = \mathbf{a} \quad \text{and} \quad \bar{\mathbf{P}}^\top\mathbf{1}_{M+1} = \mathbf{b}. \tag{9}$$

Sinkhorn Algorithm: The solution of the above optimization problem corresponds to the optimal transport [39] between discrete distributions $\mathbf{a}$ and $\mathbf{b}$ with scores $\bar{\mathbf{S}}$. Its entropy-regularized formulation naturally results in the desired soft assignment, and can be efficiently solved on GPU

with the Sinkhorn algorithm [55, 12]. It is a differentiable version of the Hungarian algorithm [35], classically used for bipartite matching, that consists in iteratively normalizing $\exp(\bar{\mathbf{S}})$ along rows and columns, similar to row and column Softmax. After T iterations, we drop the dustbins and recover $\mathbf{P} = \bar{\mathbf{P}}_{1:M,1:N}$.

3.3. Loss

By design, both the graph neural network and the optimal matching layer are differentiable – this enables backpropagation from matches to visual descriptors. SuperGlue is trained in a supervised manner from ground truth matches $\mathcal{M} = \{(i,j)\} \subset \mathcal{A} \times \mathcal{B}$. These are estimated from ground truth relative transformations – using poses and depth maps or homographies. This also lets us label some keypoints $\mathcal{I} \subseteq \mathcal{A}$ and $\mathcal{J} \subseteq \mathcal{B}$ as unmatched if they do not have any reprojection in their vicinity. Given these labels, we minimize the negative log-likelihood of the assignment $\bar{\mathbf{P}}$:

$$\begin{aligned} \text{Loss} = &- \sum_{(i,j)\in\mathcal{M}} \log \bar{\mathbf{P}}_{i,j} \\ &- \sum_{i\in\mathcal{I}} \log \bar{\mathbf{P}}_{i,N+1} - \sum_{j\in\mathcal{J}} \log \bar{\mathbf{P}}_{M+1,j}. \end{aligned} \tag{10}$$

This supervision aims at simultaneously maximizing the precision and the recall of the matching.

3.4. Comparisons to related work

The SuperGlue architecture is equivariant to permutation of the keypoints within an image. Unlike other handcrafted or learned approaches, it is also equivariant to permutation *of the images*, which better reflects the symmetry of the problem and provides a beneficial inductive bias. Additionally, the optimal transport formulation enforces reciprocity of the matches, like the mutual check, but in a soft manner, similar to [46], thus embedding it into the training process.

SuperGlue vs. Instance Normalization [60]: Attention, as used by SuperGlue, is a more flexible and powerful context aggregation mechanism than instance normalization, which treats all keypoints equally, as used by previous work on feature matching [33, 71, 32, 44, 7].

SuperGlue vs. ContextDesc [32]: SuperGlue can jointly reason about appearance and position while ContextDesc processes them separately. Moreover, ContextDesc is a front-end that additionally requires a larger regional extractor, and a loss for keypoints scoring. SuperGlue only needs local features, learned or handcrafted, and can thus be a simple drop-in replacement for existing matchers.

SuperGlue vs. Transformer [61]: SuperGlue borrows the self-attention from the Transformer, but embeds it into a graph neural network, and additionally introduces the cross-attention, which is symmetric. This simplifies the architecture and results in better feature reuse across layers.

4. Implementation details

SuperGlue can be combined with any local feature detector and descriptor but works particularly well with SuperPoint [18], which produces repeatable and sparse keypoints – enabling very efficient matching. Visual descriptors are bilinearly sampled from the semi-dense feature map. For a fair comparison to other matchers, unless explicitly mentioned, we do not train the visual descriptor network when training SuperGlue. At test time, one can use a confidence threshold (we choose 0.2) to retain some matches from the soft assignment, or use all of them and their confidence in a subsequent step, such as weighted pose estimation.

Architecture details: All intermediate representations (key, query value, descriptors) have the same dimension $D = 256$ as the SuperPoint descriptors. We use $L = 9$ layers of alternating multi-head self- and cross-attention with 4 heads each, and perform $T = 100$ Sinkhorn iterations. The model is implemented in PyTorch [38], contains 12M parameters, and runs in real-time on an NVIDIA GTX 1080 GPU: a forward pass takes on average **69 ms (15 FPS)** for an indoor image pair (see Appendix C).

Training details: To allow for data augmentation, SuperPoint detect and describe steps are performed on-the-fly as batches during training. A number of random keypoints are further added for efficient batching and increased robustness. More details are provided in Appendix E.

5. Experiments

5.1. Homography estimation

We perform a large-scale homography estimation experiment using real images and synthetic homographies with both robust (RANSAC) and non-robust (DLT) estimators.

Dataset: We generate image pairs by sampling random homographies and applying random photometric distortions to real images, following a recipe similar to [16, 18, 45, 44]. The underlying images come from the set of 1M distractor images in the Oxford and Paris dataset [42], split into training, validation, and test sets.

Local features	Matcher	Homography estimation AUC		P	R
		RANSAC	DLT		
SuperPoint	NN	39.47	0.00	21.7	65.4
	NN + mutual	42.45	0.24	43.8	56.5
	NN + PointCN	43.02	45.40	76.2	64.2
	NN + OANet	44.55	52.29	82.8	64.7
	SuperGlue	**53.67**	**65.85**	**90.7**	**98.3**

Table 1: **Homography estimation.** SuperGlue recovers almost all possible matches while suppressing most outliers. Because SuperGlue correspondences are high-quality, the Direct Linear Transform (DLT), a least-squares based solution with no robustness mechanism, outperforms RANSAC.

Baselines: We compare SuperGlue against several matchers applied to SuperPoint local features – the Nearest Neighbor (NN) matcher and various outlier rejectors: the mutual NN constraint, PointCN [33], and Order-Aware Network (OANet) [71]. All learned methods, including SuperGlue, are trained on ground truth correspondences, found by projecting keypoints from one image to the other. We generate homographies and photometric distortions on-the-fly – an image pair is never seen twice during training.

Metrics: Match precision (P) and recall (R) are computed from the ground truth correspondences. Homography estimation is performed with both RANSAC and the Direct Linear Transformation [24] (DLT), which has a direct least-squares solution. We compute the mean reprojection error of the four corners of the image and report the area under the cumulative error curve (AUC) up to a value of 10 pixels.

Results: SuperGlue is sufficiently expressive to master homographies, achieving 98% recall and high precision (see Table 1). The estimated correspondences are so good that a robust estimator is not required – SuperGlue works even better with DLT than RANSAC. Outlier rejection methods like PointCN and OANet cannot predict more correct matches than the NN matcher itself, overly relying on the initial descriptors (see Figure 6 and Appendix A).

5.2. Indoor pose estimation

Indoor image matching is very challenging due to the lack of texture, the abundance of self-similarities, the complex 3D geometry of scenes, and large viewpoint changes. As we show in the following, SuperGlue can effectively learn priors to overcome these challenges.

Dataset: We use ScanNet [13], a large-scale indoor dataset composed of monocular sequences with ground truth poses and depth images, and well-defined training, validation, and test splits corresponding to different scenes. Previous works select training and evaluation pairs based on time difference [37, 17] or SfM covisibility [33, 71, 7], usually computed using SIFT. We argue that this limits the difficulty of the pairs, and instead select these based on an overlap score computed for all possible image pairs in a given sequence using only ground truth poses and depth. This results in significantly wider-baseline pairs, which corresponds to the current frontier for real-world indoor image matching. Discarding pairs with too small or too large overlap, we select 230M training and 1500 test pairs.

Metrics: As in previous work [33, 71, 7], we report the AUC of the pose error at the thresholds $(5^\circ, 10^\circ, 20^\circ)$, where the pose error is the maximum of the angular errors in rotation and translation. Relative poses are obtained from essential matrix estimation with RANSAC. We also report the match precision and the matching score [18, 69], where a match is deemed correct based on its epipolar distance.

Local features	Matcher	Pose estimation AUC @5°	@10°	@20°	P	MS
ORB	NN + GMS	5.21	13.65	25.36	72.0	5.7
D2-Net	NN + mutual	5.25	14.53	27.96	46.7	12.0
ContextDesc	NN + ratio test	6.64	15.01	25.75	51.2	9.2
SIFT	NN + ratio test	5.83	13.06	22.47	40.3	1.0
	NN + NG-RANSAC	6.19	13.80	23.73	61.9	0.7
	NN + OANet	6.00	14.33	25.90	38.6	4.2
	SuperGlue	**6.71**	**15.70**	**28.67**	**74.2**	**9.8**
SuperPoint	NN + mutual	9.43	21.53	36.40	50.4	18.8
	NN + distance + mutual	9.82	22.42	36.83	63.9	14.6
	NN + GMS	8.39	18.96	31.56	50.3	19.0
	NN + PointCN	11.40	25.47	41.41	71.8	25.5
	NN + OANet	11.76	26.90	43.85	74.0	25.7
	SuperGlue	**16.16**	**33.81**	**51.84**	**84.4**	**31.5**

Table 2: **Wide-baseline indoor pose estimation.** We report the AUC of the pose error, the matching score (MS) and precision (P), all in percents %. SuperGlue outperforms all handcrafted and learned matchers when applied to both SIFT and SuperPoint.

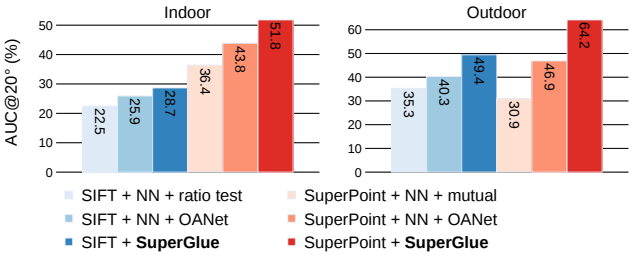

Figure 5: **Indoor and outdoor pose estimation.** SuperGlue works with SIFT or SuperPoint local features and consistently improves by a large margin the pose accuracy over OANet, a state-of-the-art outlier rejection neural network.

Baselines: We evaluate SuperGlue and various baseline matchers using both root-normalized SIFT [31, 2] and SuperPoint [18] features. SuperGlue is trained with correspondences and unmatched keypoints derived from ground truth poses and depth. All baselines are based on the Nearest Neighbor (NN) matcher and potentially an outlier rejection method. In the "Handcrafted" category, we consider the mutual check, the ratio test [31], thresholding by descriptor distance, and the more complex GMS [6]. Methods in the "Learned" category are PointCN [33], and its follow-ups OANet [71] and NG-RANSAC [7]. We retrain PointCN and OANet on ScanNet for both SuperPoint and SIFT with the classification loss using the above-defined correctness criterion and their respective regression losses. For NG-RANSAC, we use the original trained model. We do not include any graph matching methods as they are orders of magnitude too slow for the number of keypoints that we consider (>500). Other local features are evaluated as reference: ORB [47] with GMS, D2-Net [19], and ContextDesc [32] using the publicly available trained models.

Results: SuperGlue enables significantly higher pose accuracy compared to both handcrafted and learned matchers (see Table 2 and Figure 5), and works well with both SIFT and SuperPoint. It has a significantly higher precision than other learned matchers, demonstrating its higher representation power. It also produces a larger number of correct matches – up to 10 times more than the ratio test when applied to SIFT, because it operates on the full set of possible matches, rather than the limited set of nearest neighbors. SuperGlue with SuperPoint achieves state-of-the-art results on indoor pose estimation. They complement each other well since repeatable keypoints make it possible to estimate a larger number of correct matches even in very challenging situations (see Figure 2, Figure 6, and Appendix A).

5.3. Outdoor pose estimation

As outdoor image sequences present their own set of challenges (e.g., lighting changes and occlusion), we train and evaluate SuperGlue for pose estimation in an outdoor setting. We use the same evaluation metrics and baseline methods as in the indoor pose estimation task.

Dataset: We evaluate on the PhotoTourism dataset, which is part of the CVPR'19 Image Matching Challenge [1]. It is a subset of the YFCC100M dataset [56] and has ground truth poses and sparse 3D models obtained from an off-the-shelf SfM tool [37, 52, 53]. All learned methods are trained on the larger MegaDepth dataset [29], which also has depth maps computed with multi-view stereo. Scenes that are in the PhotoTourism test set are removed from the training set. Similarly as in the indoor case, we select challenging image pairs for training and evaluation using an overlap score computed from the SfM covisibility as in [19, 37].

Results: As shown in Table 3, SuperGlue outperforms all baselines, at all relative pose thresholds, when applied to both SuperPoint and SIFT. Most notably, the precision of the resulting matching is very high (84.9%), reinforcing the analogy that SuperGlue "glues" together local features.

Local features	Matcher	Pose estimation AUC @5°	@10°	@20°	P	MS
ContextDesc	NN + ratio test	20.16	31.65	44.05	56.2	3.3
SIFT	NN + ratio test	15.19	24.72	35.30	43.4	1.7
	NN + NG-RANSAC	15.61	25.28	35.87	64.4	1.9
	NN + OANet	18.02	28.76	40.31	55.0	3.7
	SuperGlue	**23.68**	**36.44**	**49.44**	**74.1**	**7.2**
SuperPoint	NN + mutual	9.80	18.99	30.88	22.5	4.9
	NN + GMS	13.96	24.58	36.53	47.1	4.7
	NN + OANet	21.03	34.08	46.88	52.4	8.4
	SuperGlue	**34.18**	**50.32**	**64.16**	**84.9**	**11.1**

Table 3: **Outdoor pose estimation.** Matching SuperPoint and SIFT features with SuperGlue results in significantly higher pose accuracy (AUC), precision (P), and matching score (MS) than with handcrafted or other learned methods.

Matcher		Pose AUC@20°	Match precision	Matching score
NN + mutual		36.40	50.4	18.8
SuperGlue	No Graph Neural Net	38.56	66.0	17.2
	No cross-attention	42.57	74.0	25.3
	No positional encoding	47.12	75.8	26.6
	Smaller (3 layers)	46.93	79.9	30.0
	Full (9 layers)	**51.84**	**84.4**	**31.5**

Table 4: **Ablation of SuperGlue.** While the optimal matching layer alone improves over the baseline Nearest Neighbor matcher, the Graph Neural Network explains the majority of the gains brought by SuperGlue. Both cross-attention and positional encoding are critical for strong gluing, and a deeper network further improves the precision.

5.4. Understanding SuperGlue

Ablation study: To evaluate our design decisions, we repeat the indoor experiments with SuperPoint features, but this time focusing on different SuperGlue variants. This ablation study, presented in Table 4, shows that all SuperGlue blocks are useful and bring substantial performance gains.

When we additionally backpropagate through the SuperPoint descriptor network while training SuperGlue, we observe an improvement in AUC@20°from 51.84 to 53.38. This confirms that SuperGlue is suitable for end-to-end learning beyond matching.

Visualizing Attention: The extensive diversity of self- and cross-attention patterns is shown in Figure 7 and reflects the complexity of the learned behavior. A detailed analysis of the trends and inner-workings is performed in Appendix D.

6. Conclusion

This paper demonstrates the power of attention-based graph neural networks for local feature matching. SuperGlue's architecture uses two kinds of attention: (i) self-attention, which boosts the receptive field of local descriptors, and (ii) cross-attention, which enables cross-image communication and is inspired by the way humans look back-and-forth when matching images. Our method elegantly handles partial assignments and occluded points by solving an optimal transport problem. Our experiments show that SuperGlue achieves significant improvement over existing approaches, enabling highly accurate relative pose estimation on extreme wide-baseline indoor and outdoor image pairs. In addition, SuperGlue runs in real-time and works well with both classical and learned features.

In summary, our learnable middle-end replaces handcrafted heuristics with a powerful neural model that simultaneously performs context aggregation, matching, and filtering in a single unified architecture. We believe that, when combined with a deep front-end, SuperGlue is a major milestone towards end-to-end deep SLAM.

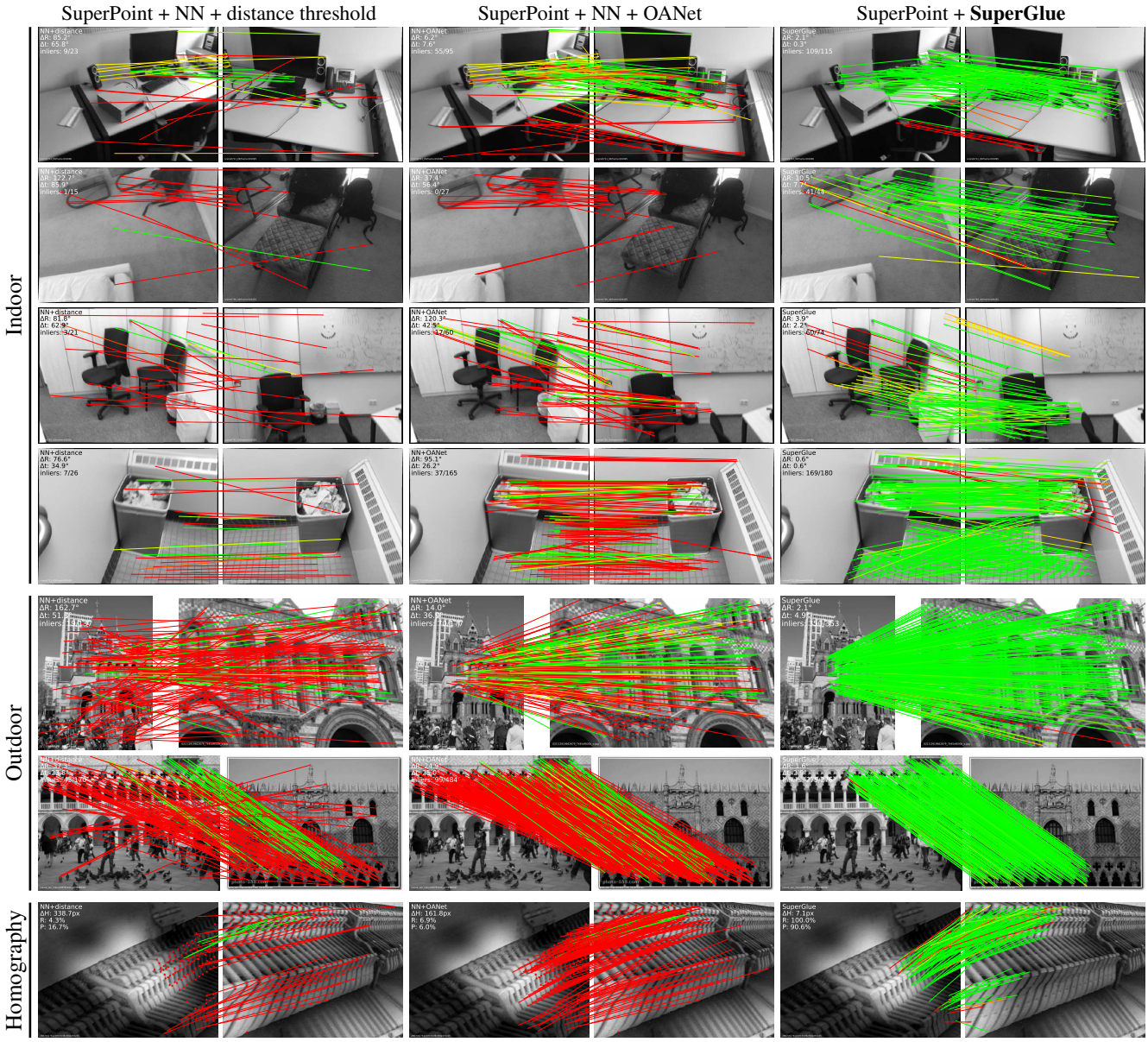

Figure 6: **Qualitative image matches.** We compare SuperGlue to the Nearest Neighbor (NN) matcher with two outlier rejectors, handcrafted and learned, in three environments. SuperGlue consistently estimates more correct matches (green lines) and fewer mismatches (red lines), successfully coping with repeated texture, large viewpoint, and illumination changes.

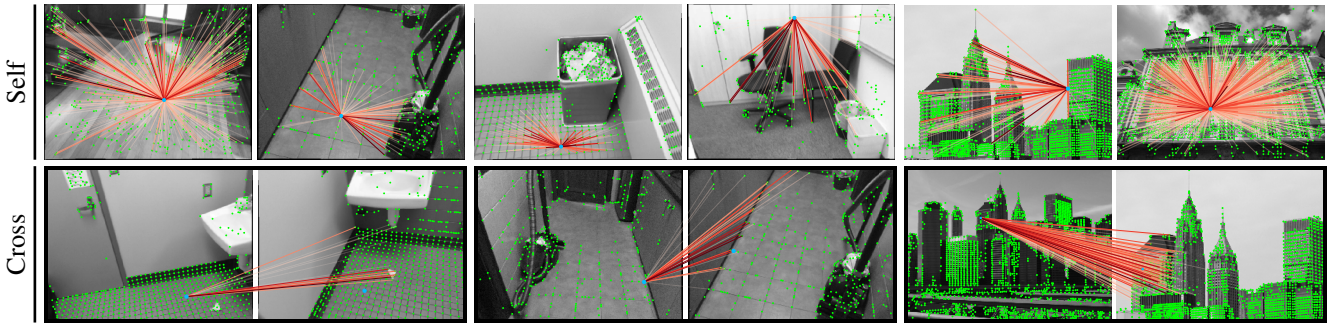

Figure 7: **Visualizing attention.** We show self- and cross-attention weights α_{ij} at various layers and heads. SuperGlue exhibits a diversity of patterns: it can focus on global or local context, self-similarities, distinctive features, or match candidates.

Appendix

In the following pages, we present additional experimental details, quantitative results, qualitative examples of SuperGlue in action, detailed timing results, as well as visualizations and analysis of the learned attention patterns.

A. Detailed results

A.1. Homography estimation

Qualitative results: A full page of qualitative results of SuperGlue matching on synthetic and real homographies can be seen in Figure 13.

Synthetic dataset: We take a more detailed look at the homography evaluation from Section 5.1. Figure 8 shows the match precision at several correctness pixel thresholds and the cumulative error curve of homography estimation. SuperGlue dominates across all pixel correctness thresholds.

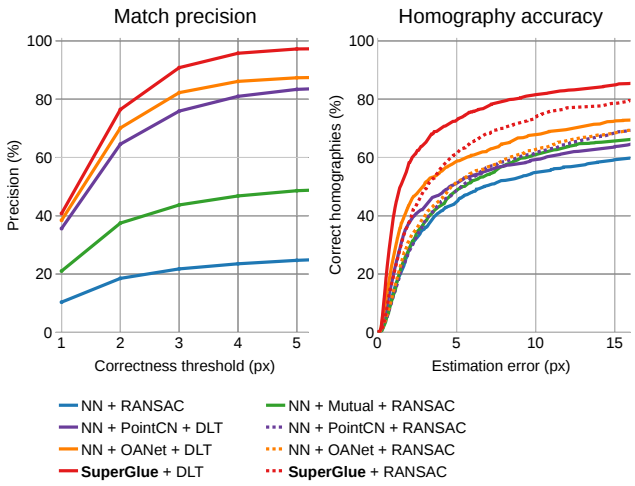

Figure 8: **Details of the homography evaluation.** SuperGlue exhibits higher precision and homography accuracy at all thresholds. High precision results in more accurate estimation with DLT than with RANSAC.

Local features	Matcher	Viewpoint		Illumination	
		P	R	P	R
SuperPoint	NN	39.7	81.7	51.1	84.9
	NN + mutual	65.6	77.1	74.2	80.7
	NN + PointCN	87.6	80.7	94.5	82.6
	NN + OANet	90.4	81.2	**96.3**	83.5
	SuperGlue	**91.4**	**95.7**	89.1	**91.7**

Table 5: **Generalization to real data.** We show the precision (P) and recall (R) of the methods trained on our synthetic homography dataset (see Section 5.1) on the viewpoint and illumination subsets of the HPatches dataset. While trained on synthetic homographies, SuperGlue generalizes well to real data.

HPatches: We assess the generalization ability of SuperGlue on real data with the HPatches [3] dataset, as done in previous works [18, 45]. This dataset depicts planar scenes with ground truth homographies and contains 295 image pairs with viewpoint changes and 285 pairs with illumination changes. We evaluate the models trained on the synthetic dataset (see Section 5.1). The HPatches experiment is summarized in Table 5. As previously observed in the synthetic homography experiments, SuperGlue has significantly higher recall than all matchers relying on the NN search. We attribute the remaining gap in recall to several challenging pairs for which SuperPoint does not detect enough repeatable keypoints. Nevertheless, synthetic-dataset trained SuperGlue generalizes well to real data.

A.2. Indoor pose estimation

Qualitative results: More visualizations of matches computed by SuperGlue on indoor images are shown in Figure 14, and highlight the extreme difficulty of the wide-baseline image pairs that constitute our evaluation dataset.

ScanNet: We present more details regarding the results on ScanNet (Section 5.2), only analyzing the methods which use SuperPoint local features. Figure 9 plots the cumulative pose estimation error curve and the trade-off between precision and number of correct matches. We compute the correctness from the reprojection error (using the ground truth depth and a threshold of 10 pixels), and, for keypoints with invalid depth, from the symmetric epipolar error. We obtain curves by varying the confidence thresholds of PointCN, OANet, and SuperGlue. At evaluation, we use the original value 0.5 for the former two, and 0.2 for SuperGlue.

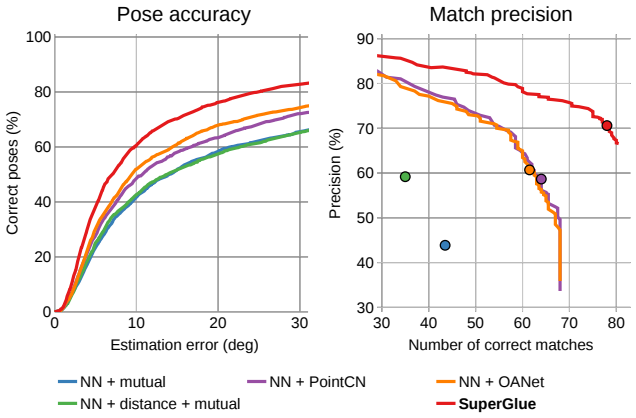

Figure 9: **Details of the ScanNet evaluation.** Poses estimated with SuperGlue are more accurate at all error thresholds. SuperGlue offers the best trade-off between precision and number of correct matches, which are both critical for accurate and robust pose estimation.

Local features	Matcher	Exact AUC			Approx. AUC [71]		
		5°	10°	20°	5°	10°	20°
ContextDesc	NN + ratio test	26.09	45.52	63.07	53.00	63.13	73.00
SIFT	NN + ratio test	24.09	40.71	58.14	45.12	55.81	67.20
	NN + OANet*	28.76	48.42	66.18	55.50	65.94	76.17
	NN + OANet	29.15	48.12	65.08	55.06	64.97	74.83
	SuperGlue	**30.49**	**51.29**	**69.72**	**59.25**	**70.38**	**80.44**
SuperPoint	NN + mutual	16.94	30.39	45.72	35.00	43.12	54.05
	NN + OANet	26.82	45.04	62.17	50.94	61.41	71.77
	SuperGlue	**38.72**	**59.13**	**75.81**	**67.75**	**77.41**	**85.70**

Table 6: **Outdoor pose estimation on YFCC100M pairs.** The evaluation is performed on the same image pairs as in OANet [71] using both their approximate and our exact AUC. SuperGlue consistently improves over the baselines when using either SIFT and SuperPoint.

A.3. Outdoor pose estimation

Qualitative results: Figure 15 shows additional results on the Phototourism test set and the MegaDepth validation set.

YFCC100M: While the PhotoTourism [1] and Zhang *et al.*'s [71] test sets are both based on YFCC100M [56], they use different scenes and pairs. For the sake of comparability, we also evaluate SuperGlue on the same evaluation pairs as in OANet [71], using their evaluation metrics. We include an OANet model (*) retrained on their training set (instead of MegaDepth) using root-normalized SIFT. The results are shown in Table 6.

As observed in Section 5.3 when evaluating on the PhotoTourism dataset, SuperGlue consistently improves over all baselines for both SIFT and SuperPoint. For SIFT, the improvement over OANet is decreased, which we attribute to the significantly higher overlap and lower difficulty of the pairs used by [71]. While the approximate AUC tends to overestimate the accuracy, it results in an identical ranking of the methods. The numbers for OANet with SIFT and SuperPoint are consistent with the ones reported in their paper.

Method	Correctly localized queries (%)			# features
	.5m/2°	1m/5°	5m/10°	
R2D2 [45]	**46.9**	66.3	**88.8**	20k
D2-Net [19]	45.9	68.4	**88.8**	15k
UR2KID [68]	**46.9**	67.3	**88.8**	15k
SuperPoint+NN+mutual	43.9	59.2	76.5	4k
SuperPoint+SuperGlue	45.9	**70.4**	**88.8**	4k

Table 7: **Visual localization on Aachen Day-Night.** SuperGlue significantly improves the performance of SuperPoint for localization, reaching new state-of-the-art results with comparably fewer keypoints.

B. SuperGlue for visual localization

Visual localization: While two-view relative pose estimation is an important fundamental problem, advances in image matching can directly benefit practical tasks like visual localization [50, 48], which aims at estimating the absolute pose of a query image with respect to a 3D model. Moreover, real-world localization scenarios exhibit significantly higher scene diversity and more challenging conditions, such as larger viewpoint and illumination changes, than phototourism datasets of popular landmarks.

Evaluation: The Aachen Day-Night benchmark [51, 50] evaluates local feature matching for day-night localization. We extract up to 4096 keypoints per images with SuperPoint, match them with SuperGlue, triangulate an SfM model from posed day-time database images, and register night-time query images with the 2D-2D matches and COLMAP [52]. The evaluation server[1] computes the percentage of queries localized within several distance and orientation thresholds. As reported in Table 7, SuperPoint+SuperGlue performs similarly or better than all existing approaches despite using significantly fewer keypoints. Figure 10 shows challenging day-night image pairs.

[1] https://www.visuallocalization.net/

Figure 10: **Matching challenging day-night pairs with SuperGlue.** We show predicted correspondences between night-time queries and day-time databases images of the Aachen Day-Night dataset. The correspondences are colored as RANSAC inliers in green or outliers in red. Although the outdoor training set has few night images, SuperGlue generalizes well to such extreme illumination changes. Moreover, it can accurately match building facades with repeated patterns like windows.

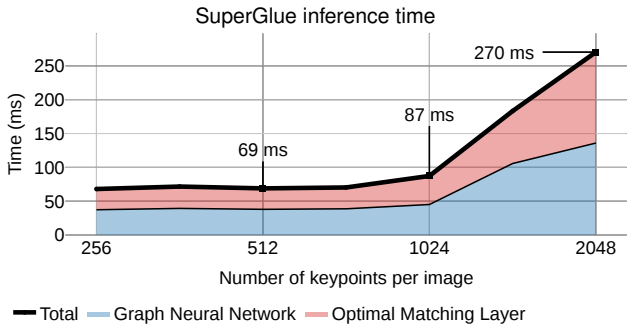

Figure 11: **SuperGlue detailed inference time.** SuperGlue's two main blocks, the Graph Neural Network and the Optimal Matching Layer, have similar computational costs. For 512 and 1024 keypoints per image, SuperGlue runs at 14.5 and 11.5 FPS, respectively.

C. Timing and model parameters

Timing: We measure the run-time of SuperGlue and its two major blocks, the Graph Neural Network and the Optimal Matching Layer, for different numbers of keypoints per image. The measurements are performed on an NVIDIA GeForce GTX 1080 GPU across 500 runs. See Figure 11.

Model Parameters: The Keypoint Encoder MLP has 5 layers, mapping positions to dimensions of size $(32, 64, 128, 256, D)$, yielding 100k parameters. Each layer has the three projection matrices, and an extra $\mathbf{W}^O$ to deal with the multi-head output. The message update MLP has 2 layers and maps to dimensions $(2D, D)$. Both MLPs use BatchNorm and ReLUs. Each layer has 0.66M parameters. SuperGlue has 18 layers, with a total of 12M parameters.

D. Analyzing attention

Quantitative analysis: We compute the spatial extent of the attention weights – the *attention span* – for all layers and all keypoints. The self-attention span corresponds to the distance in pixel space between one keypoint i and all the others j, weighted by the attention weight α_{ij}, and averaged for all queries. The cross-attention span corresponds to the average distance between the final predicted match and all the attended keypoints j. We average the spans over 100 ScanNet pairs and plot in Figure 12 the minimum across all heads for each layer, with 95% confidence intervals.

The spans of both self- and cross-attention tend to decrease throughout the layers, by more than a factor of 10 between the first and the last layer. SuperGlue initially attends to keypoints covering a large area of the image, and later focuses on specific locations – the self-attention attends to a small neighborhood around the keypoint, while the cross-attention narrows its search to the vicinity of the true match. Intermediate layers have oscillating spans, hinting at a more complex process.

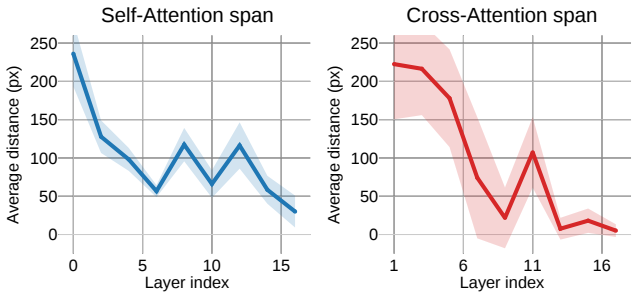

Figure 12: **Attention spans throughout SuperGlue.** We plot the attention span, a measure of the attention's spatial dispersion, vs. layer index. For both types of attention, the span tends to decrease deeper in the network as SuperGlue focuses on specific locations. See an example in Figure 16.

Qualitative example: We analyze the attention patterns of a specific example in Figure 16. Our observations are consistent with the attention span trends reported in Figure 12.

E. Experimental details

In this section, we provide details on the training and evaluation of SuperGlue. The trained models and the evaluation code and image pairs are publicly available at `github.com/magicleap/SuperGluePretrainedNetwork`.

Choice of indoor dataset: Previous works on inlier classification [33, 71, 7] evaluate indoor pose estimation on the SUN3D dataset [67]. Camera poses in SUN3D are estimated from SIFT-based sparse SfM, while ScanNet leverages RGB-D fusion and optimization [14], resulting in significantly more accurate poses. This makes ScanNet more suitable for generating accurate correspondence labels and evaluating pose estimation. We additionally noticed that the SUN3D image pairs used by Zhang *et al.* [71] have generally small baseline and rotation angle. This makes the essential matrix estimation degenerate [24] and the angular translation error ill-defined. In contrast, our ScanNet wide-baseline pairs have significantly more diversity in baselines and rotation, and thus do not suffer from the aforementioned issues.

Homography estimation – Section 5.1: The test set contains 1024 pairs of 640×480 images. Homographies are generated by applying random perspective, scaling, rotation, and translation to the original full-sized images, to avoid bordering artifacts. We evaluate with the 512 top-scoring keypoints detected by SuperPoint with a Non-Maximum Suppression (NMS) radius of 4 pixels. Correspondences are deemed correct if they have a reprojection error lower than 3 pixels. We use the OpenCV function `findHomography` with 3000 iterations and a RANSAC inlier threshold of 3 pixels.

Indoor pose estimation – Section 5.2: The overlap score between two images A and B is the average ratio of pixels in A that are visible in B (and vice versa), after accounting for missing depth values and occlusion (by checking for consistency in the depth). We train and evaluate with pairs that have an overlap score in $[0.4, 0.8]$. For training, we sample at each epoch 200 pairs per scene, similarly as in [19]. The test set is generated by subsampling the sequences by 15 and subsequently randomly sampling 15 pairs for each of the 300 sequences. We resize all ScanNet images and depth maps to 640×480. We detect up to 1024 SuperPoint keypoints (using the publicly available trained model[2] with NMS radius of 4) and 2048 SIFT keypoints (using OpenCV's implementation). Poses are computed by first estimating the essential matrix with OpenCV's `findEssentialMat` and RANSAC with an inlier threshold of 1 pixel divided by the focal length, followed by `recoverPose`. In contrast with previous works [33, 71, 7], we compute a more accurate AUC using explicit integration rather than coarse histograms. The precision (P) is the average ratio of the number of correct matches over the total number of estimated matches. The matching score (MS) is the average ratio of the number of correct matches over the total number of detected keypoints. It does not account for the pair overlap and decreases with the number of covisible keypoints. A match is deemed correct if its epipolar distance is lower than $5 \cdot 10^{-4}$.

Outdoor pose estimation – Section 5.3: For training on Megadepth, the overlap score is the ratio of triangulated keypoints that are visible in the two images, as in [19]. We sample pairs with an overlap score in $[0.1, 0.7]$ at each epoch. We evaluate on all 11 scenes of the PhotoTourism dataset and reuse the overlap score based on bounding boxes computed by Ono *et al*. [37], with a selection range of $[0.1, 0.4]$. Images are resized so that their longest dimension is equal to 1600 pixels and rotated upright using their EXIF data. We detect 2048 keypoints for both SIFT and SuperPoint (with an NMS radius of 3). The epipolar correctness threshold is here 10^{-4}. Other evaluation parameters are identical to the ones used for the indoor evaluation.

Training of SuperGlue: For training on homography/indoor/outdoor data, we use the Adam optimizer [25] with a constant leaning rate of 10^{-4} for the first 200k/100k/50k iterations, followed by an exponential decay of 0.999998/0.999992/0.999992 until iteration 900k. When using SuperPoint features, we employ batches with 32/64/16 image pairs and a fixed number of 512/400/1024 keypoints per image. For SIFT features we use 1024 keypoints and 24 pairs. Due to the limited number of training scenes, the outdoor model weights are initialized with the homography model weights. Before the keypoint encoder, the keypoints are normalized by the largest dimension of the image.

Ground truth correspondences $\mathcal{M}$ and unmatched sets $\mathcal{I}$ and $\mathcal{J}$ are generated by first computing the $M \times N$ reprojection matrix between all detected keypoints using the ground truth homography or pose and depth. Correspondences are entries with a reprojection error that is a minimum along both rows and columns, and that is lower than a given threshold: 3, 5, and 3 pixels for homographies, indoor, and outdoor matching respectively. For homographies, unmatched keypoints are simply the ones that do not appear in $\mathcal{M}$. For indoor and outdoor matching, because of errors in the pose and depth, unmatched keypoints must additionally have a minimum reprojection error larger than 15 and 5 pixels, respectively. This allows us to ignore labels for keypoints whose correspondences are ambiguous, while still providing some supervision through the normalization induced by the Sinkhorn algorithm.

Ablation study – Section 5.4: The "No Graph Neural Net" baseline replaces the Graph Neural Network with a single linear projection, but retains the Keypoint Encoder and the Optimal Matching Layer. The "No cross-attention" baseline replace all cross-attention layers by self-attention: it has the same number of parameters as the full model, and acts like a Siamese network. The "No positional encoding" baseline simply removes the Keypoint Encoder and only uses the visual descriptors as input.

End-to-end training – Section 5.4: Two copies of SuperPoint, for detection and description, are initialized with the original weights. The detection network is frozen and gradients are propagated through the descriptor network only, flowing from SuperGlue - no additional losses are used.

[2] github.com/magicleap/SuperPointPretrainedNetwork

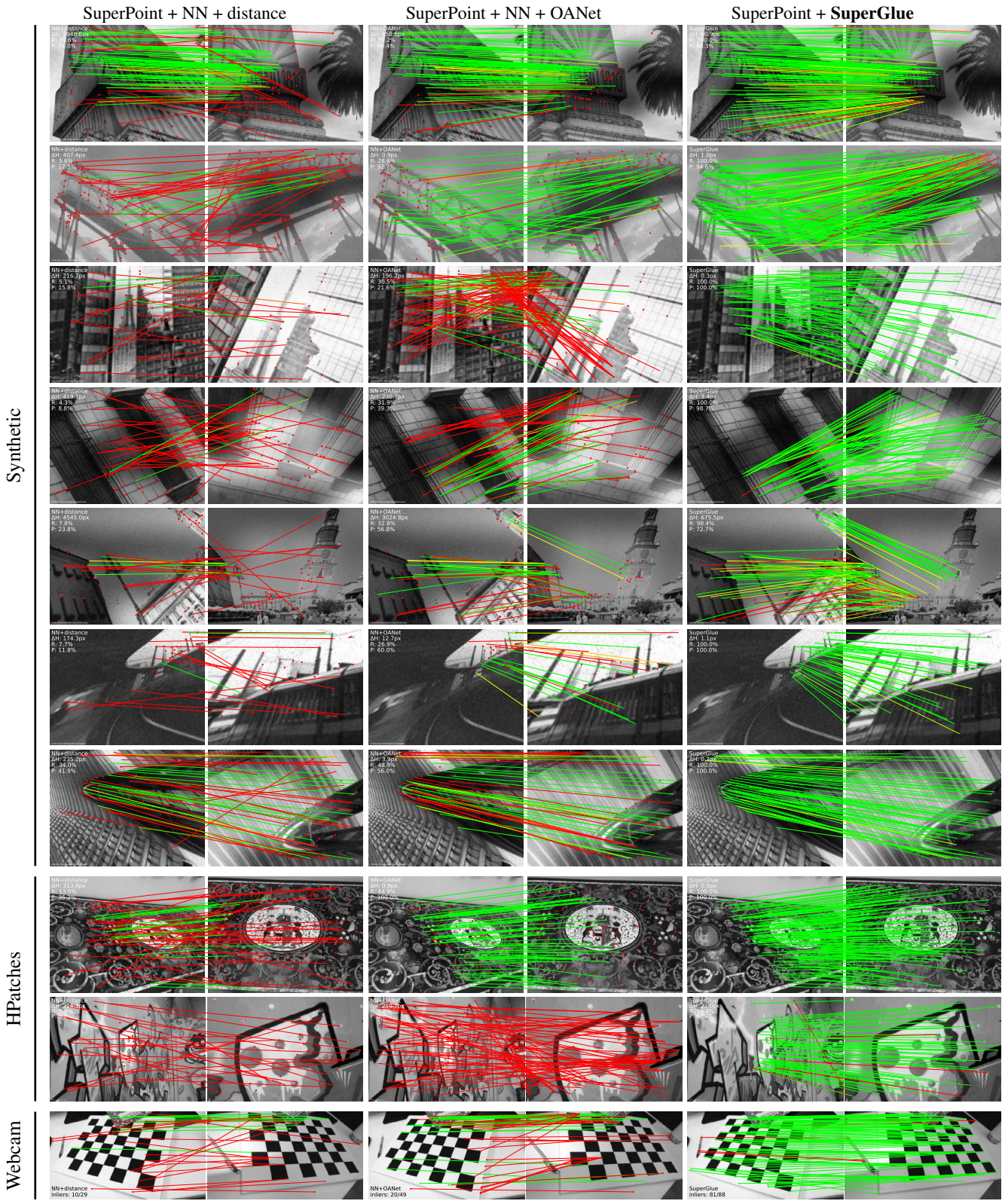

Figure 13: **More homography examples.** We show point correspondences on our synthetic dataset (see Section 5.1), on real image pairs from HPatches (see Appendix A.1), and a checkerboard image captured by a webcam. SuperGlue consistently estimates more correct matches (green lines) and fewer mismatches (red lines), successfully coping with repeated texture, large viewpoint, and illumination changes.

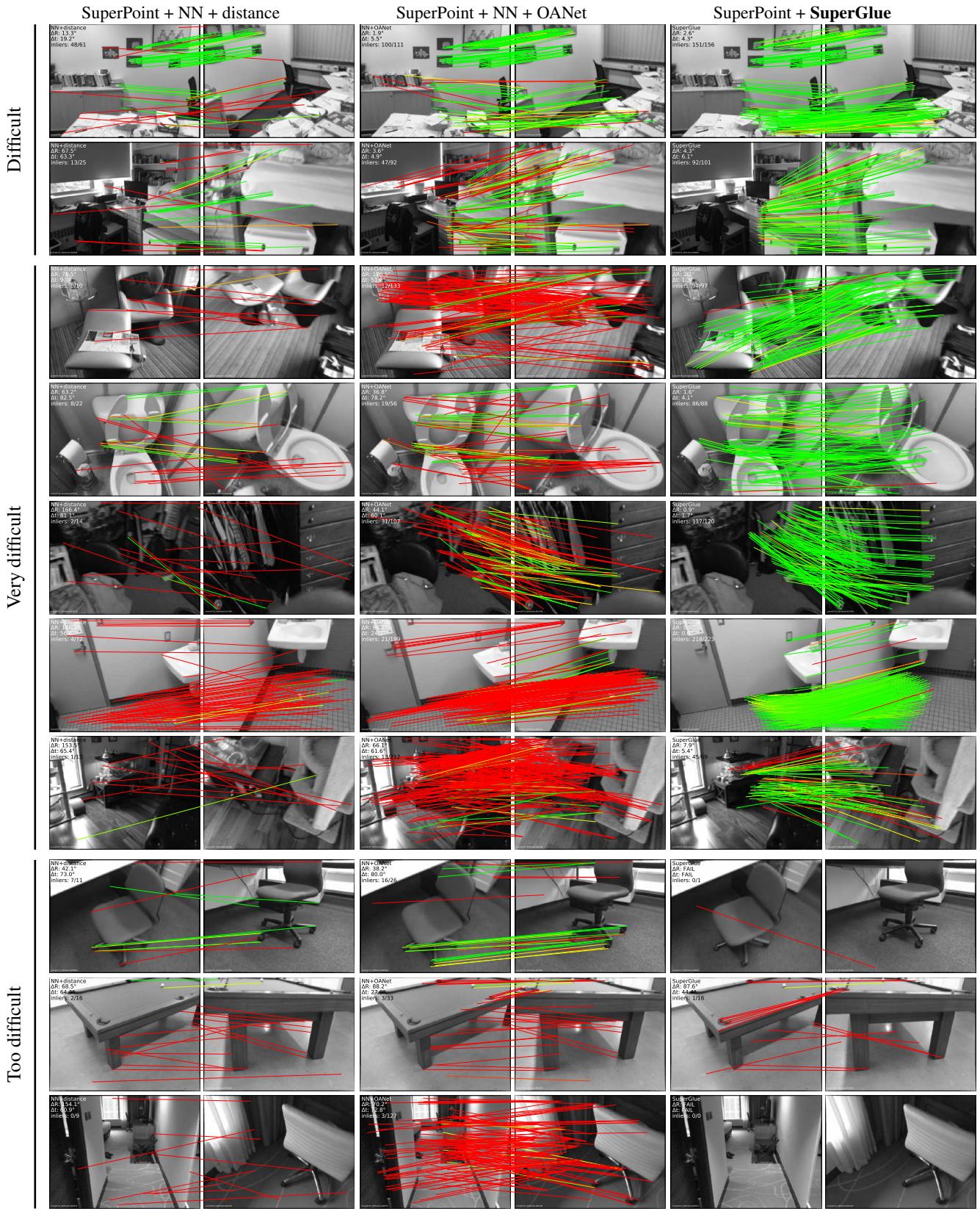

Figure 14: **More indoor examples.** We show both **Difficult** and **Very Difficult** ScanNet indoor examples for which SuperGlue works well, and three **Too Difficult** examples where it fails, either due to unlikely motion or lack of repeatable keypoints (last two rows). Correct matches are green lines and mismatches are red lines. See details in Section 5.2.

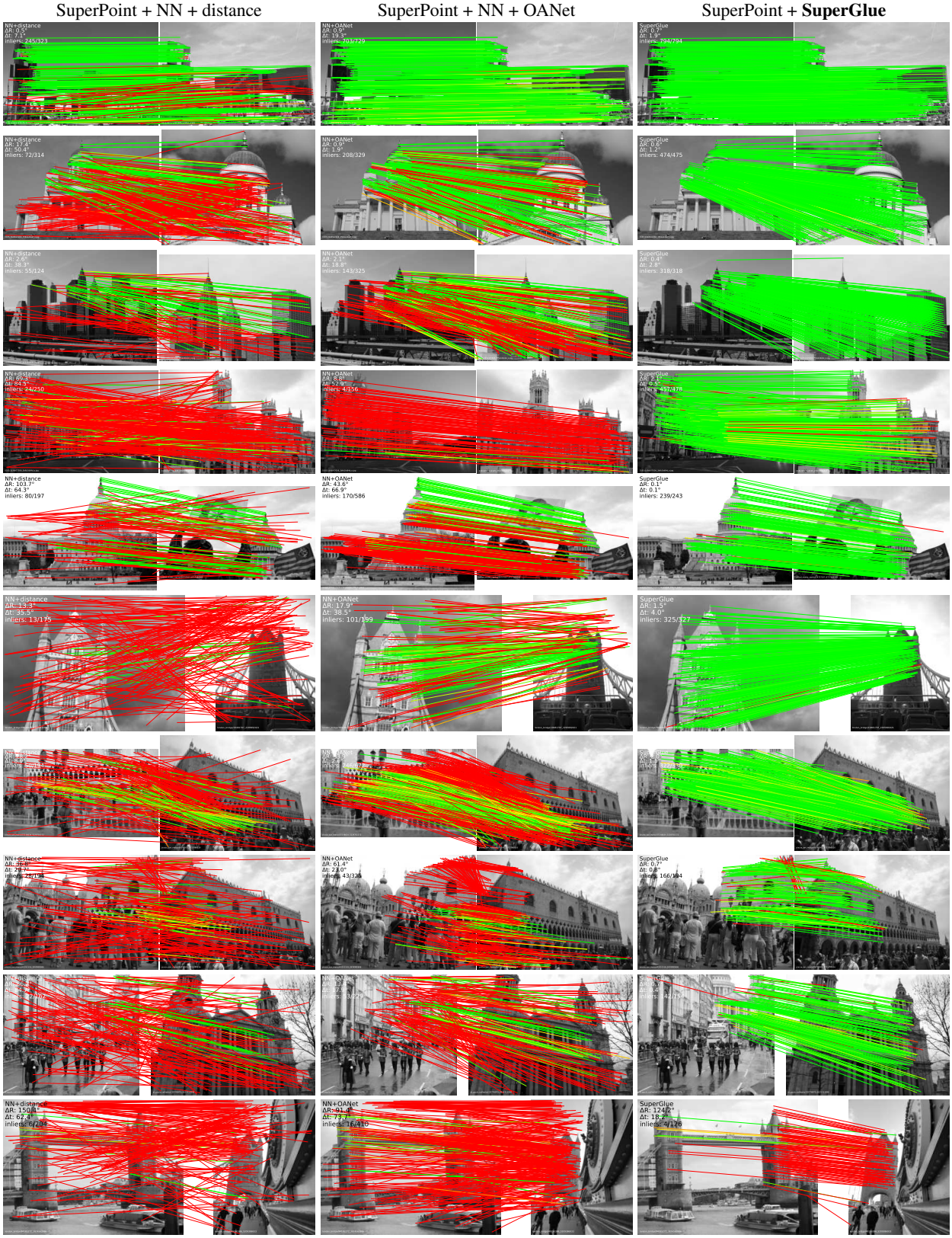

Figure 15: **More outdoor examples.** We show results on the MegaDepth validation and the PhotoTourism test sets. Correct matches are green lines and mismatches are red lines. The last row shows a failure case, where SuperGlue focuses on the incorrect self-similarity. See details in Section 5.3.

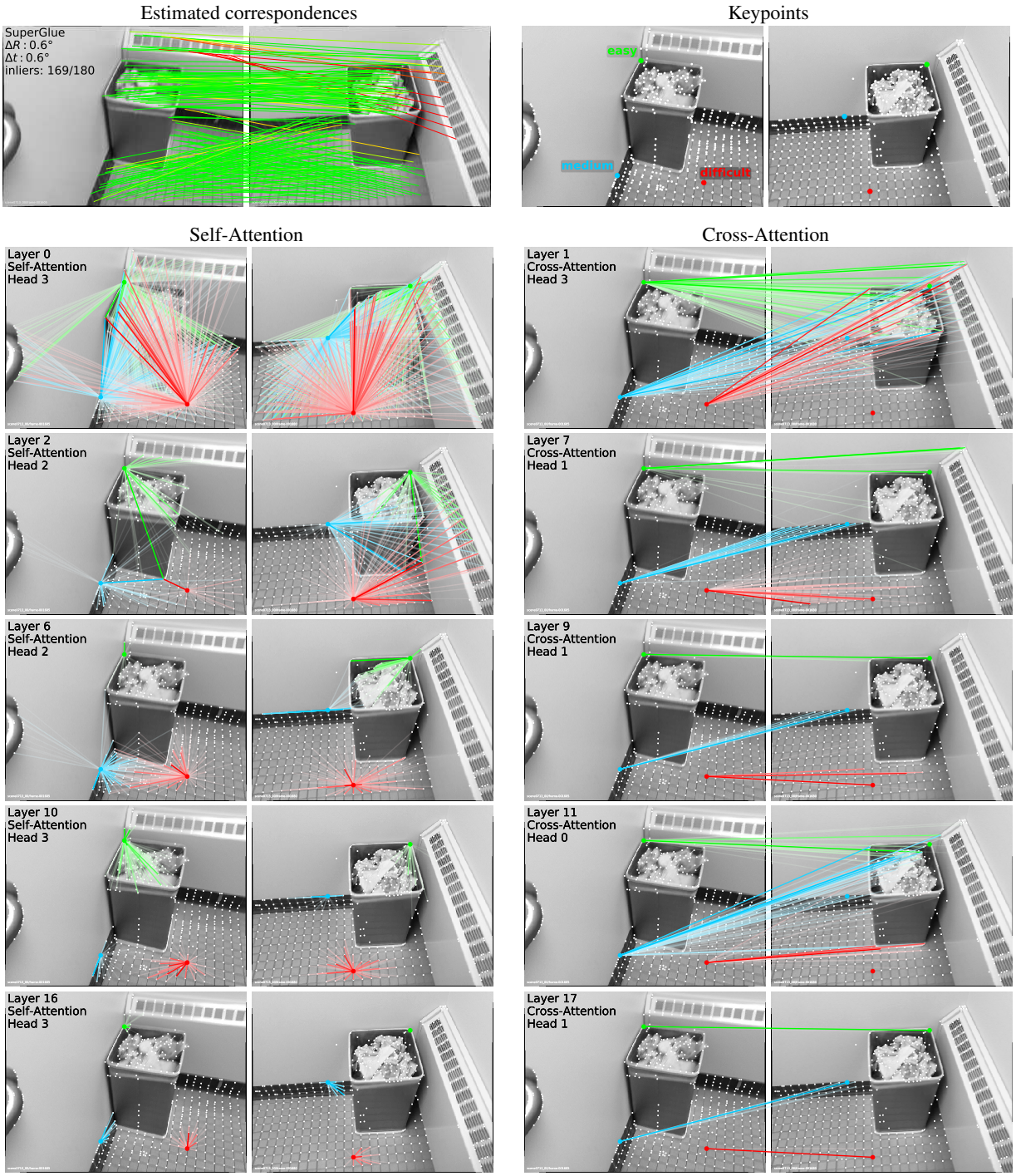

Figure 16: **Attention patterns across layers.** For this image pair (correctly matched by SuperGlue), we look at three specific keypoints that can be matched with different levels of difficulty: the easy keypoint, the medium keypoint, and the difficult keypoint. We visualize self- and cross-attention weights (within images A and B, and from A to B, respectively) of selected layers and heads, varying the edge opacity with α_{ij}. The self-attention initially attends all over the image (row 1), and gradually focuses on a small neighborhood around each keypoint (last row). Similarly, some cross-attention heads focus on candidate matches, and successively reduce the set that is inspected. The easy keypoint is matched as early as layer 9, while more difficult ones are only matched at the last layer. Similarly as in Figure 12, the self- and cross-attention spans generally shrink throughout the layers. They however increase in layer 11, which attends to other locations – seemingly distinctive ones – that are further away. We hypothesize that SuperGlue attempts to disambiguate challenging matches using additional context.

References

[1] Phototourism Challenge, CVPR 2019 Image Matching Workshop. https://image-matching-workshop.github.io. Accessed November 8, 2019. 7, 10

[2] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012. 6

[3] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017. 9

[4] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv:1806.01261*, 2018. 2, 3

[5] Alexander C Berg, Tamara L Berg, and Jitendra Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR*, 2005. 2

[6] JiaWang Bian, Wen-Yan Lin, Yasuyuki Matsushita, Sai-Kit Yeung, Tan-Dat Nguyen, and Ming-Ming Cheng. GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *CVPR*, 2017. 2, 6

[7] Eric Brachmann and Carsten Rother. Neural-Guided RANSAC: Learning where to sample model hypotheses. In *ICCV*, 2019. 2, 5, 6, 11, 12

[8] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016. 1

[9] Tibério S Caetano, Julian J McAuley, Li Cheng, Quoc V Le, and Alex J Smola. Learning graph matching. *IEEE TPAMI*, 31(6):1048–1058, 2009. 2

[10] Jan Cech, Jiri Matas, and Michal Perdoch. Efficient sequential correspondence selection by cosegmentation. *IEEE TPAMI*, 32(9):1568–1581, 2010. 2

[11] Marvin M Chun. Contextual cueing of visual attention. *Trends in cognitive sciences*, 4(5):170–178, 2000. 3

[12] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, 2013. 2, 5

[13] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 6

[14] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(3):24, 2017. 11

[15] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPFNet: Global context aware local features for robust 3D point matching. In *CVPR*, 2018. 2

[16] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. In *RSS Workshop: Limits and Potentials of Deep Learning in Robotics*, 2016. 5

[17] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Self-improving visual odometry. *arXiv:1812.03245*, 2018. 6

[18] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *CVPR Workshop on Deep Learning for Visual SLAM*, 2018. 2, 4, 5, 6, 9

[19] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable CNN for joint detection and description of local features. In *CVPR*, 2019. 2, 6, 7, 10, 12

[20] Patrick Ebel, Anastasiia Mishchuk, Kwang Moo Yi, Pascal Fua, and Eduard Trulls. Beyond cartesian representations for local descriptors. In *ICCV*, 2019. 2

[21] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[22] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *ICML*, 2017. 3

[23] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017. 2, 3

[24] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 6, 11

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 12

[26] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, 2019. 2

[27] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 2005. 2

[28] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. In *ICML*, 2019. 2

[29] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 7

[30] Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey for the quadratic assignment problem. *European journal of operational research*, 176(2):657–690, 2007. 2

[31] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2, 6

[32] Zixin Luo, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. ContextDesc: Local descriptor augmentation with cross-modality context. In *CVPR*, 2019. 2, 3, 5, 6

[33] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *CVPR*, 2018. 2, 5, 6, 11, 12

[34] Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010. 3

[35] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957. 5

[36] Vincenzo Nicosia, Ginestra Bianconi, Vito Latora, and Marc Barthelemy. Growing multiplex networks. *Physical review letters*, 111(5):058701, 2013. 3

[37] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning local features from images. In *NeurIPS*, 2018. 2, 6, 7, 12

[38] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Workshops*, 2017. 5

[39] Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. 2, 4

[40] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 2

[41] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 2

[42] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting Oxford and Paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018. 5

[43] 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

[44] René Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *ECCV*, 2018. 2, 5

[45] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2D2: Repeatable and reliable detector and descriptor. In *NeurIPS*, 2019. 2, 5, 9, 10

[46] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *NeurIPS*, 2018. 2, 5

[47] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, 2011. 6

[48] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 10

[49] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. SCRAMSAC: Improving RANSAC's efficiency with a spatial consistency filter. In *ICCV*, 2009. 2

[50] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *CVPR*, 2018. 10

[51] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012. 10

[52] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 7, 10

[53] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 7

[54] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007. 3

[55] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 1967. 2, 5

[56] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 7, 10

[57] Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV*, 2008. 2

[58] Tomasz Trzcinski, Jacek Komorowski, Lukasz Dabala, Konrad Czarnota, Grzegorz Kurzejamski, and Simon Lynen. SConE: Siamese constellation embedding descriptor for image matching. In *ECCV Workshops*, 2018. 3

[59] Tinne Tuytelaars and Luc J Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *BMVC*, 2000. 2

[60] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv:1607.08022*, 2016. 2, 5

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 1, 2, 3, 4, 5

[62] Petar Velikovi, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Li, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. 2

[63] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008. 2

[64] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 2

[65] Yue Wang and Justin M Solomon. Deep Closest Point: Learning representations for point cloud registration. In *ICCV*, 2019. 2

[66] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic Graph CNN for learning on point clouds. *ACM Transactions on Graphics*, 2019. 2

[67] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. SUN3D: A database of big spaces reconstructed using SfM and object labels. In *ICCV*, 2013. 11

[68] Tsun-Yi Yang, Duy-Kien Nguyen, Huub Heijnen, and Vassileios Balntas. UR2KiD: Unifying retrieval, keypoint detection, and keypoint description without local correspondence supervision. *arXiv:2001.07252*, 2020. 10

[69] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned invariant feature transform. In *ECCV*, 2016. 2, 6

[70] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *NIPS*, 2017. 2

[71] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *ICCV*, 2019. 2, 5, 6, 10, 11, 12

[72] Li Zhang, Xiangtai Li, Anurag Arnab, Kuiyuan Yang, Yunhai Tong, and Philip HS Torr. Dual graph convolutional network for semantic segmentation. In *BMVC*, 2019. 2

[73] Yimeng Zhang, Zhaoyin Jia, and Tsuhan Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR*, 2011. 3