Consensus Learning with Deep Sets for Essential Matrix Estimation

Dror Moran   **Yuval Margalit**   **Guy Trostianetsky**

Fadi Khatib   **Meirav Galun**   **Ronen Basri**

Department of Computer Science and Applied Mathematics
Weizmann Institute of Science

Abstract

Robust estimation of the essential matrix, which encodes the relative position and orientation of two cameras, is a fundamental step in structure from motion pipelines. Recent deep-based methods achieved accurate estimation by using complex network architectures that involve graphs, attention layers, and hard pruning steps. Here, we propose a simpler network architecture based on Deep Sets. Given a collection of point matches extracted from two images, our method identifies outlier point matches and models the displacement noise in inlier matches. A weighted DLT module uses these predictions to regress the essential matrix. Our network achieves accurate recovery that is superior to existing networks with significantly more complex architectures.

1 Introduction

Estimating the relative pose of two cameras depicting a stationary scene is a fundamental computer vision task and a basic step in multiview structure from motion (SFM) [33, 24, 45, 1, 51, 43, 20, 26, 35] and simultaneous localization and mapping (SLAM) [22, 47, 7, 21] pipelines. Both classical and recent deep network-based algorithms (see a review in Section 2) use point matches to compute the essential matrix, which encodes the relative position and orientation of the two cameras. Identifying such point matches by existing heuristics, however, is prone to mistakes, due to possibly large viewpoint changes, illumination differences, and the presence of ambiguous repetitive scene structures, resulting in *noisy matches* and extremely large numbers of *outlier matches* (often as many as 95%) that must be identified and pruned to enable accurate pose recovery.

Classical SFM algorithms use RANSAC [16] to robustly identify inliers and estimate pose parameters. While RANSAC has been used effectively for consensus recovery, learning-based deep network approaches have introduced a competitive alternative, making steady progress in accuracy while allowing for efficient inference and demonstrating resilience to very large fractions of outliers. This progress was obtained at the price of complicating the network architecture, e.g., using message passing in local, near-neighbour graphs [58, 28, 31, 52] or expensive attention (transformer) layers [28, 52], along with the addition of hard pruning steps [58, 28, 52].

In this paper, we introduce a simpler network architecture for consensus learning based on the Deep Sets framework [55]. Deep Sets architectures are based on shared, element-wise layers that are combined with global features produced by summing the element-wise features. Zaheer et al. and others [55, 50] proved that such architectures can express universal permutation-equivariant functions over sets. In our network, the input set elements include pairs of *keypoints*, i.e., the coordinates of matching pairs of points. In each layer, element-wise features are produced by a linear layer

with shared weights, followed by SoftPlus activation. Global features are obtained by averaging the element-wise features, where averaging is used to maintain invariance to set cardinality. The network utilizes a stack of such permutation equivariant layers to classify point matches as either inliers or outliers and identify a consensus set to enable accurate relative camera pose regression. We further improve accuracy by integrating a noise regression module that aims to predict the displacement, due to noise, of the (clean) positions of inlier keypoints. Finally, we observe that training in two stages, i.e., first on a noise-free version of the real data (while including the outliers) and subsequently on the original real data, improves the accuracy of the predicted pose. Our network achieves accurate pose recovery that is superior to existing networks with significantly more complex architectures.

In summary, our contributions include:

- NACNet, a Noise Aware Consensus Network, for consensus learning tasks and robust geometric model estimation.
- A DeepSets based architecture that includes inlier displacement error estimation.
- An effective noise-free pretraining scheme: first, pretrain on a denoised version of the real data, then train on the real (noisy) data.
- Experiments demonstrate that NACNet achieves superior results compared to baselines on indoor and outdoor image pairs applied on various descriptors.

Our code is available at `https://github.com/drormoran/NACNet`.

2   Related work

Classical methods. RANSAC [16] and its successors, LO-RANSAC [8], USAC [39], MAGSAC [3], and MAGSAC++ [4] search over minimal point configurations to find consensus sets from noisy and corrupted data and estimate a corresponding parametric model. These are applied to matched keypoints with distinct descriptors obtained by filtering with Lowe's ratio test [30]. These classical methods are regarded as the standard solutions for finding consensus in data consisting of mixtures of inlier and outlier point matches.

Learning-based methods. Deep learning-based methods have been used recently to regress a geometric model and outlier classification. DFE [40] used a deep-based iteratively reweighted least squares (IRLS) scheme to predict inlier/outlier scores. LFGC [34] utilized an architecture that involves an inlier/outlier classifier and weight sharing, followed by context normalization, and applied a geometric loss to the output of the weighted 8-point algorithm (also called weighted DLT [17]).

Follow-up works improve prediction results by introducing more complex network designs. OANet [56] introduced an order-aware block, which contains differentiable permutation invariant pooling and unpooling operators that capture local context by utilizing soft clustering of correspondences in the feature space. CLNet [58] used this order-aware block together with pruning and local-to-global consensus learning procedure strategy to classify the correspondences by employing convolutions on local and global graphs built based on the Euclidean distance in feature space. All the methods mentioned above suffer from the leakage of outliers to the consensus set. Consequently, they all use RANSAC at the end of their inference step. In contrast, NCMNet [28], MGNet [31], and BCLNet [52] used weighted DLT also at inference, showing that the performance of the results is not improved further when RANSAC is applied in addition. NCMNet [28] proposed a local-to-global consensus learning scheme in which it first creates a local spatial graph, then a local feature space graph, and finally a global graph based on the inlier scores from the local graphs. BCLNet [52] introduced the idea of Bilateral Consensus, adopting the local graph from CLNet [58] as their projection step in a channel-wise transformer that learns global consensus. MGNet [31] used a similar scheme, building both implicit and explicit local graphs and a global graph. Unlike previous methods, this method does not prune correspondences inside the network.

In contrast to these methods, we use an architecture based on Deep Sets [55]. Deep Sets enable efficient information transfer between the point matches through global features without the need to construct and manipulate graphs. Our newly proposed noise regression module further improves our results. Finally, as with recent methods, our method too does not require a final RANSAC step.

Learned feature matching. Deep learning-based detectors and descriptors [11, 54, 13, 14] based on both CNNs and Transformers have been used in recent years to replace the handcrafted features
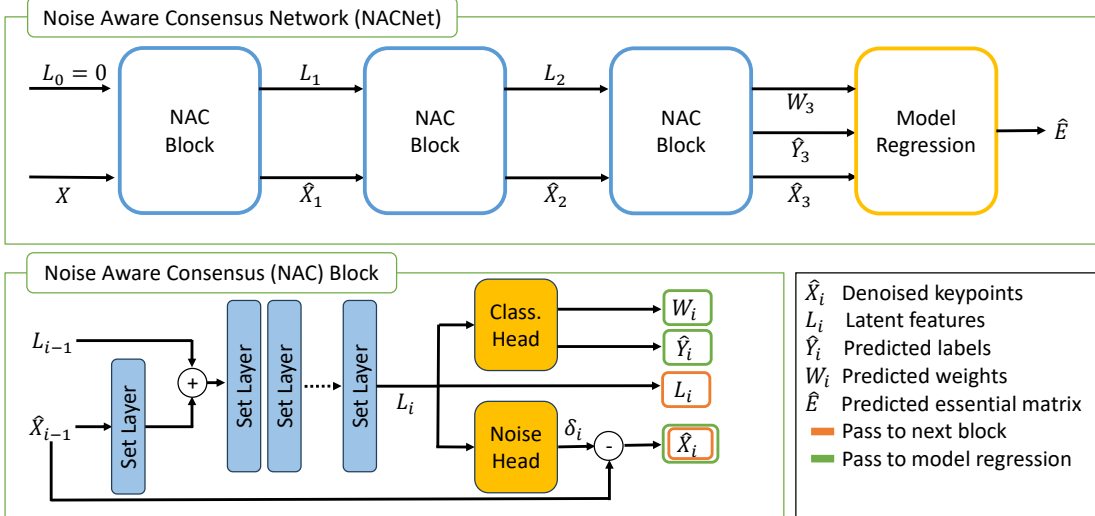
Figure 1: **Network architecture**. Noise Aware Consensus Network (NACNet) architecture, see text for details.

[30, 36, 5] used in classical methods. Those methods trained with challenging and diverse data have improved the accuracy and robustness of matching even with the classical nearest neighbor matching. Yet the main problems of high outlier rate remained. Consequently, learned matchers that match keypoints while rejecting non-matchable ones have merged[42, 27], combining Transformers with optimal transport[37] to produce more accurate matches even with large camera movement. These matchers rely on the descriptors for the matching and keypoint rejection and require RANSAC as a post-processing step. In contrast, our method, similar to [28], [31], [52], gets as an input the keypoints (point correspondences) only and does not incorporate RANSAC.

Keypoint refinement. Previous works [15, 48, 12, 26] have shown that correcting keypoints position could positively influence the results of geometric model estimation. All of those works use visual and learned features (SIFT descriptors or features obtained from applying a convolutional network to the input images) to correct the positions. To our knowledge, our paper is the first to use the input point set directly to correct keypoint position, as opposed to previous works, which rely on either an estimated geometric model or visual features.

3 Method

Consider a pair of images captured by (internally) calibrated cameras expressed with 3×4 matrices, $P = [I, \mathbf{0}]$ and $\tilde{P} = [R, \mathbf{t}]$, where R and $\mathbf{t}$ respectively denote the relative rotation and translation between the two views. The essential matrix $E = [\mathbf{t}]_\times R$, determines the epipolar geometry between the two views, so that for any two corresponding points, $\mathbf{p}$ and $\tilde{\mathbf{p}}$, projected from a 3D point, it holds that $\tilde{\mathbf{p}}^T E \mathbf{p} = 0$. Existing algorithms commonly estimate the essential matrix directly from a set of putative matches between the two views, i.e., pairs of keypoints.

Our aim in this work is to construct a network that identifies a consensus set of point matches (a set of inliers), given a set of putative matches as input (generally contaminated with noise and outliers), and, based on this consensus set, predicts the essential matrix between the two images. We seek to construct a network that can overcome positional noise, which can reside in the inlier matches, and cope with a considerable fraction of outlier matches, up and above 95%. In addition, we aim for a method that can generalize to unseen image pairs and work with a varying number of point matches and a variety of fractions of outliers.

Those goals are achieved by employing a permutation-equivariant network architecture with the following key properties: (1) a two-stage noise-aware training scheme, (2) a noise head for predicting positional inlier noise, and (3) a classification head to discriminate between inliers and outliers. These three key properties are at the core of our method. Hence, we refer to our network as a Noise-Aware Consensus Network (NACNet).
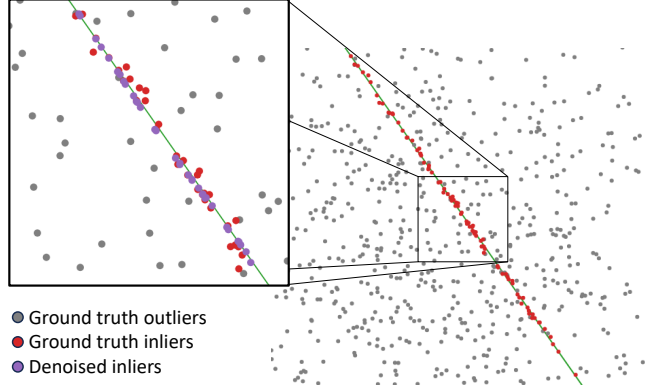
Figure 2: **NACNet point location denoising on a line-fitting task.** The set X (right panel) is composed of 90% outliers (marked in grey) and (noisy) inliers (red). Our model predicts the denoised version $\hat{X}$ (purple, left panel). Evidently, the prediction of the positional noise, yielding noise-free inliers, agrees with the line model.

Formally, let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^4$ denote a set of point correspondences, with $\mathbf{x}_i = (p_i, q_i, \tilde{p}_i, \tilde{q}_i)$ denoting a match between a keypoint $\mathbf{p}_i = (p_i, q_i)$ in the left image and a keypoint $\tilde{\mathbf{p}}_i = (\tilde{p}_i, \tilde{q}_i)$ in the right image. Our aim is, given X as an input to classify each matching pair $\mathbf{x}_i$ as an inlier ($y_i = 1$) or outlier ($y_i = 0$) and associate with it an (inlier) confidence score $C_i \in \mathbb{R}$. Those predictions are used to estimate the essential matrix that relates the two views.

Network architecture. Our network comprises of three noise-aware consensus (NAC) blocks. Each NAC block uses a set encoder to map the input matches to a latent representation and to correct their positions due to positional noise. The third block further classifies the points as inliers or outliers and produces their corresponding confidence scores. Its outputs feed the model regression block, which implements a weighted differentiable Direct Linear Transformation (DLT) algorithm [34], based on the confidence scores, to predict the essential matrix. We refer the reader to Figure 1 for a detailed scheme of our network architecture.

Noise aware consensus (NAC) block. Each NAC block comprises of a set encoder, a noise head, and a classification head. The set encoder uses DeepSets layers to map the coordinates of the input point matches to a latent representation $L \in \mathbb{R}^{n \times d}$. Each DeepSets layer includes a linear, permutation equivariant layer followed by SoftPlus activation. These layers apply a linear transformation to each set member and an additional (different) linear transformation to their average. (We replace the sum in [55] with an average to maintain invariance to set cardinality.)

The noise and classification heads are implemented with simple two-layer MLPs. The noise head uses the latent representation to predict displacement vectors for all input points, $\delta \in \mathbb{R}^{n \times 4}$. These displacement vectors are subtracted from the input points, X, producing their predicted denoised locations, $\hat{X}$. The classification head uses as input the latent representation and outputs predictions for the inlier/outlier classification labels $\hat{Y} \in [0, 1]^n$ with their corresponding weights $W \in \mathbb{R}^n$.

The predicted denoised version of the keypoints $\hat{X}$ and the latent representation L are passed to the next NAC block. In the third block, the inlier/outlier predicted labels $\hat{Y}$, the weights W, and the denoised keypoints $\hat{X}$ are passed to the model regression block.

We demonstrate the NAC block denoising effect on a simple line-fitting task. We randomly sample 100 noisy points on a line and, in addition, 900 outliers. An example is shown in Figure 2, where our NACNet significantly reduces the positional noise in the inlier points.

Model regression block. The model regression block uses the predicted weights, W, and the classification labels, $\hat{Y}$, obtained from the classification head, and the denoised version of the keypoints, $\hat{X}$, obtained from the noise head to predict the essential matrix in the following way.

$$\hat{E} = g(\hat{X}, \hat{Y}, W), \tag{1}$$

where g denotes the differentiable DLT algorithm (also called the weighted eight-point algorithm, see formulation in [34], Section 3). Similarly to [46], we calculate confidence scores as follows

$$C_i = \frac{\hat{Y}_i \cdot \exp(W_i)}{\sum_j \hat{Y}_j \cdot \exp(W_j)}. \tag{2}$$

The confidence scores are used as the weights corresponding to the denoised keypoints $\hat{X}$ in the weighted DLT algorithm.

3.1 Loss function

We minimize a loss composed of three terms

$$L(\hat{X}, \hat{Y}, \hat{E}; X, Y, E) = L_{\text{cls}}(\hat{Y}, Y) + \alpha_{\text{mod}} L_{\text{mod}}(\hat{E}, E) + \alpha_{\text{ns}} L_{\text{ns}}(\hat{X}, X, E). \tag{3}$$

The first term L_{cls} uses a weighted binary cross entropy loss, due to the imbalanced of the inliers and outliers in the data, to penalize for inlier/outlier classification errors

$$L_{\text{cls}}(\hat{Y}, Y) = -\frac{1}{n} \sum_{i=1}^{n} \left[\beta_{\text{inliers}} \cdot y_i \cdot \log(\hat{y}_i) + \beta_{\text{outliers}} \cdot (1 - y_i) \cdot \log(1 - \hat{y}_i) \right]. \tag{4}$$

Here, n is the cardinality of the keypoint set, X, and β_{inliers} and β_{outliers} are determined by a hyperparameter search.

The second term L_{mod} penalizes for errors in the predicted essential matrix, similarly to the suggestion in [40]. We start with a grid of k point pairs and use the Optimal Triangulation Method (OTM) [17] (page 318) to find the closest points that satisfy the ground truth epipolar constraints. Specifically, given a pair of grid points $(\mathbf{p}_i, \tilde{\mathbf{p}}_i)$, OTM seeks to find the global minimum for the following optimization problem:

$$(\mathbf{q}_i, \tilde{\mathbf{q}}_i) = \underset{\mathbf{p}_i', \tilde{\mathbf{p}}_i'}{\text{argmin}} \, d(\mathbf{p}_i, \mathbf{p}_i')^2 + d(\tilde{\mathbf{p}}_i, \tilde{\mathbf{p}}_i')^2 \quad \text{subject to } \tilde{\mathbf{p}}_i'^T E \mathbf{p}_i' = 0, \tag{5}$$

where E is the ground truth essential matrix, and d is the Euclidean distance between the points. Given the optimal points $\{\mathbf{q}_i, \tilde{\mathbf{q}}_i\}_{i=1}^{k}$, we define the loss using the Symmetric Epipolar Distance:

$$L_{\text{mod}}(\hat{E}, E) = \sum_{i=1}^{k} (\tilde{\mathbf{q}}_i^T \hat{E} \mathbf{q}_i)^2 \left(\frac{1}{\|\hat{E}^T \tilde{\mathbf{q}}_i\|_2^2} + \frac{1}{\|\hat{E} \mathbf{q}_i\|_2^2} \right), \tag{6}$$

where $\hat{E}$ is the predicted essential matrix and $k = 400$.

The last term L_{ns} is used to minimize the distance between the noise-free keypoints, $\bar{X}$, and the predicted denoised version of the keypoints, $\hat{X}$, over the set of the ground truth inliers, as follows

$$L_{\text{ns}}(\hat{X}, X, E) = \|\hat{X}_{\text{inliers}} - \bar{X}_{\text{inliers}}\|. \tag{7}$$

To determine the ground truth, noise-free inlier keypoints, $\bar{X}_{\text{inliers}}$, we apply the Optimal Triangulation Method ([17], page 318). The parameters α_{mod} and α_{ns} are determined by a hyperparameter search.

3.2 Training

Training our model to remove outlier matches is complicated by the presence of noise in the positions of inlier matches, potentially resulting in a small classification margin. This, in turn, has been shown (in the case of kernel SVM) to have a negative effect on sample complexity and generalization error [44] (Pages 205-206, 221). A further complication is the lack of ground truth labels; i.e., our inlier/outlier labels are set by applying a preset threshold to the deviation of the points from the projections derived by the Optimal Triangulation Method ([17], page 318)(see Section 4.1).

To approach this problem, we train our model by applying a two-stage, noise-aware optimization process. The input to the first stage includes the set $\bar{X}$ containing the noise-free inlier matches along with the outlier matches. The optimization in this stage, therefore, uses only the first two terms of the loss (3), and the noise head is muted. In the second stage, the input to the network includes the original set of keypoints X, and the full loss, i.e., including (7), is optimized. Our experiments and ablations indicate that this two-stage training process significantly improves the performance of our method.
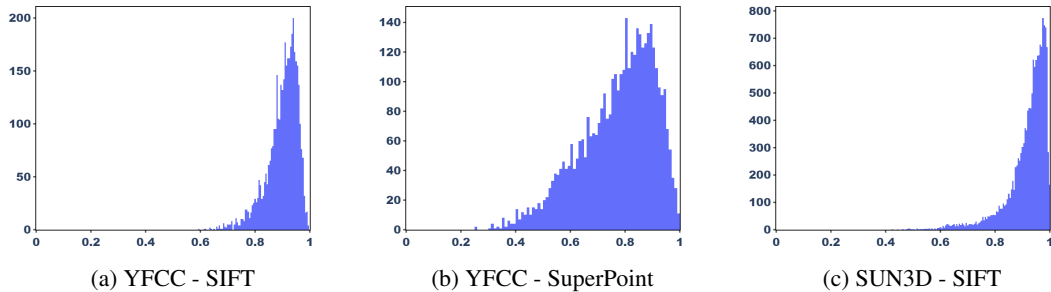
(a) YFCC - SIFT	(b) YFCC - SuperPoint	(c) SUN3D - SIFT

Figure 3: **Distributions of outliers in the different datatsets**. Histograms showing for each dataset (YFCC and Sun3D) and feature descriptor (SIFT or SuperPoint) the number of image pairs (the Y-axis) with a given fraction of outlier matches (the X-axis). The means and standard deviations (from left to right) are $0.89 \pm 0.06, 0.77 \pm 0.14, 0.92 \pm 0.08$

4 Experiments

4.1 Datasets and baselines

Datasets. We trained and tested our method on both indoor and outdoor datasets. For an outdoor dataset, we used Yahoo's YFCC dataset[49], which contains 100 million images from flicker later reconstructed using SFM[18]. For an indoor dataset, we used the SUN3D [53]. For both datasets, we used the same preprocessing and dataset split as in [56], i.e., the camera poses are extracted from an SFM pipeline, and the test set is split into in-scene and cross-scene generalization. In contrast to previous methods that use the Symmetric Epipolar Distance for "ground truth" inlier/outlier labeling, we determined the labels by the deviation of the points from the projections derived by the Optimal Triangulation Method ([17], page 318) using a threshold of 3×10^{-3}. In practice, changing the labeling paradigm did not affect the results. Additionally, we used the Phototourism dataset[19] to test our model's generalization across datasets. For keypoint detection, we used SIFT [30], ORB[36], and SuperPoint [11] followed by the preprocessing steps suggested in [58]. As is shown in Figure 3, consensus learning on these datasets is highly challenging due to the high fraction of outliers in all datasets and with all descriptors.

Baselines. We compare our methods with RANSAC[16], DEGENSAC[9], GC-RANSAC[2], MAGSAC++[4], PointNet++[38], DFE[40], LFGC[34], OA-Net[56], ACNe [32], LMC-Net[29], CL-Net[58], MS^{2}DG-Net[10], ConvMatch[57], U-Match[25], NCMNet[28], MGNet[31], BCLNet[52], and SuperGlue[42]. All the evaluations of deep learning-based methods are taken from their respective papers unless specifically stated otherwise. We used the official SuperGlue repository for evaluation on SuperPoint, and the paper[42] results for evaluation on SIFT. For the RANSAC-based methods [2, 4, 9, 16], we set the maximal number of iterations to 100K and use Lowe's ratio test[30] to filter the initial matches, with a threshold tuned differently for the SIFT and SuperPoint descriptors to maximize performance.

Evaluation metrics. We use the mean average precision (mAP) to evaluate our model predictions as suggested in [34]. We compute the mAP over the maximum between the translation and rotation angular errors of our predicted essential matrix up to the threshold of $5°$.

4.2 Essential matrix estimation

Our results are shown in Table 1-3. (Additional evaluations are shown in the Appendix.) The results demonstrate that our model outperforms the current SOTA in almost all conditions, including with indoor (SUN3D data) and outdoor (YFCC) images, with keypoint matches obtained with SIFT and SuperPoint, in in-scene (unseen image pairs from scenes included in training), cross-scene, and even cross-dataset (PhotoTourism) experiments. Specifically, in the YFCC/SIFT task (Table 1), our model outperforms the other methods by a significant margin in the in-scene generalization task and with a smaller margin in the cross-scene generalization task. Likewise, on the SUN3D dataset, our method outperforms the other methods in both in-scene and cross-scene generalization, improving over the

previous SOTA by 3.6% in the cross-scene test. Qualitative results can be seen in Figure 4 and in Appendix A.3.
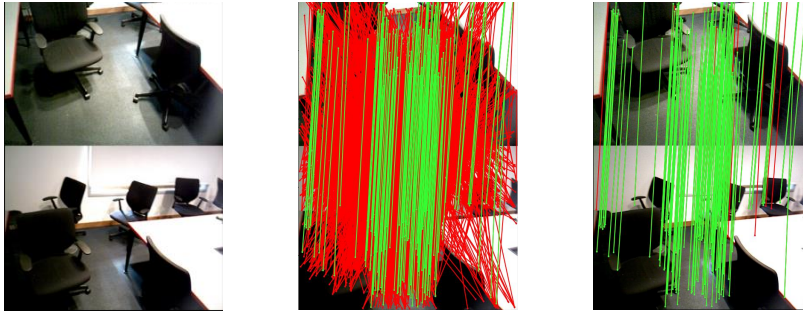

Figure 4: **NACNet inlier/outlier classification**. An example from the SUN3D dataset. Left to right: input image pairs, input matches, and our model's predicted inliers. Color mark ground truth labels: inlier matches are marked in green; outliers are marked in red.

Table 1: **SIFT evaluation**. Evaluation of essential matrix estimation on the YFCC and SUN3D datasets with keypoint matching obtained with SIFT. mAP5°(%) is reported, and the best result in each column is in bold. In-scene denote results on novel image pairs taken from scenes that were included in the training data and cross-scene denote results on image pairs taken from unseen scenes. The first set of methods (above the middle line) includes methods that incorporate RANSAC.

Method	YFCC(%)		SUN3D(%)	
	In-scene	Cross-scene	In-scene	Cross-scene
RANSAC	31.57	42.78	20.88	15.79
GC-RANSAC	30.88	42.55	18.69	13.57
LO-RANSAC	30.96	42.60	19.01	13.85
MAGSAC++	31.01	42.57	19.55	14.23
SuperGlue	-	59.25	-	-
Point-Net++	10.49	16.48	10.58	8.10
DFE	19.13	30.27	14.05	12.06
LFGC	13.81	23.95	11.55	9.30
OA-Net++	32.57	38.95	20.86	16.18
ACNe	29.17	33.06	18.86	14.12
LMC-Net	33.73	47.50	19.92	16.82
CL-Net	39.16	53.10	20.35	17.03
MS^{2}DG-Net	38.36	49.13	22.20	17.84
ConvMatch	43.48	54.62	25.36	21.71
U-Match	46.78	60.53	24.98	21.38
NCMNet	52.33	63.43	26.12	20.66
MGNet	51.43	64.63	25.96	21.27
BCLNet	52.62	66.08	24.59	19.96
Ours	**60.10**	**66.14**	**34.15**	**25.36**

We also tested our model using the deep-learning based descriptor SuperPoint on both the YFCC and SUN3D datasets. Table 2 shows that our model improves over existing SOTA methods that do not incorporate RANSAC by 6.9% on the cross-scene YFCC test and by 2.96% on the cross-scene SUN3D test. Overall, our model achieves SOTA results over both datasets and descriptors.

4.3 Generalization to different descriptors and datasets

We next examined our model's ability to generalize across different descriptors and datasets. For these experiments we used our model trained on the YFCC dataset with SIFT matches. We then tested this model on image pairs from novel scenes (i.e., cross-scene experiment) from the YFCC dataset with matches obtained using ORB and SuperPoint. Additionally, we applied the YFCC/SIFT model to image pairs from the test set of the Phototurisem dataset with matches obtained with SIFT and SuperPoint. The results are shown in Table 3. While our model slightly underperforms the SOTA with ORB matches, it outperforms existing methods with SP matches by almost 3.9%. On

Table 2: **SuperPoint evaluation**. Evaluation of camera pose estimation in experiments on outdoor and indoor datasets using matches obtained with SuperPoint. mAP5°(%) is reported, and the best result in each column is in bold. In-scene denote results on novel image pairs taken from scenes that were included in the training data and cross-scene denote results on image pairs taken from unseen scenes. The first set of methods (above the middle line) includes methods that incorporate RANSAC.

Method	YFCC(%)		SUN3D(%)	
	In-scene	Cross-scene	In-scene	Cross-scene
RANSAC	20.36	25.60	21.25	15.89
GC-RANSAC	17.27	22.27	19.32	14.17
LO-RANSAC	17.25	22.12	19.22	14.46
MAGSAC++	18.09	23.47	20.23	15.02
SuperGlue	39.71	57.45	24.09	19.45
Point-Net++	11.87	17.95	11.40	9.38
DFE	18.79	29.13	13.35	12.04
LFGC	12.18	24.25	12.63	10.68
OA-Net++	29.52	35.27	20.01	15.62
ACNe	26.72	32.98	18.35	13.82
CL-Net	29.35	38.99	15.89	14.03
MS^{2}DG-Net	30.40	37.38	20.28	16.08
U-Match	35.12	45.72	22.73	18.87
ConvMatch	38.34	48.80	25.36	21.71
NCMNet	–	52.20	–	–
MGNet	41.53	49.37	24.58	20.65
BCLNet	40.56	48.07	-	-
Ours	**55.94**	**59.10**	**33.97**	**24.67**

Table 3: **Generalization across descriptors and datasets**. This table shows the results obtained with our model trained on the YFCC dataset with SIFT matches applied in inference to the YFCC dataset with the ORB and SuperPoint (SP) feature extractors and to the PhotoTourism dataset with SIFT and SuperPoint (SP). mAP5°(%) is reported, and the best result in each column is marked in bold. † indicates evaluation conducted using published models.

	YFCC(%)		PhotoTourism(%)	
	ORB	SP	SIFT	SP
LFGC	7.40	14.78	20.17	5.89
OA-Net++	12.05	19.40	40.39	8.99
CL-Net	14.75	21.00	45.54	9.41
MS^{2}DG-Net	11.38	21.05	45.53	12.91
U-Match	16.70	28.38	54.43	11.48
NCMNet	19.95	33.20	54.73	30.60
BCLNet†	18.70	25.85	54.29	23.34
MGNet	**20.00**	32.88	57.64	20.41
Ours	19.17	**37.14**	**60.81**	**49.03**

generalization to the Phototurism dataset our method performs best with both SIFT and SuperPoint matches.

4.4 Resource utilization

In Table 4, we provide an account of the resources used by our model including the number of parameters, GPU memory usage, and runtime. It can be seen that while our model uses more parameters than NCMNet and BCLNet, which use graph attention architectures, it is 4-6 times faster than these methods and consumes less GPU memory at inference. We further compare our runtime to RANSAC-based methods. With 100K iterations, RANSAC is significantly slower than our method. We note that RANSAC is implemented in CPU. We further considered the recent Kornia's GPU implementation of RANSAC for fundamental matrix estimation [41]. Using a batch size of 10000 samples, their model runtime and maximum GPU memory usage were 40.94ms and 414.66MB, respectively, which are 4 times higher than our model. For GPU resource usage, we used an NVIDIA GeForce RTX 2080Ti and an Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz for CPU.

Table 4: **Resource utilization.** The table shows the resource usage of our model compared to previous methods in terms of number of parameters, GPU memory usage, and runtime. Tested on YFCC with SIFT descriptors. Note that GC-RANSAC and MAGASAC++ are implemented in CPU.

Methods	#Params (M)	Max GPU Mem (MB)	Runtime Avg.(ms)
GC-RANSAC	-	-	217.3
MAGASAC++	-	-	295.29
NCMNet	**4.77**	174.52	67.43
BCLNet	4.87	140.91	46.94
Ours	22.14	**130.75**	**11.12**

4.5 Ablation study

Keypoint denoising. In ablation studies we tested the importance of our noise head, i.e., keypoint denoising process, by training our model with and without this head. In both cases, we used our two stage training scheme. The results in Table 5 demonstrate that our keypoint denoising improves our model performance in both indoor (SUN3D) and outdoor (YFCC) scenes and using both SIFT and SuperPoint matches. This improvement is more noticeable in the more challenging indoor scenario, as it includes fewer inliers and less accurate positions of keypoints.

Two-stage noise-aware training. To test the importance of our two-stage training scheme, we trained our model in a single stage on the original (noisy) set of keypoints X, while the noise head is muted on the YFCC dataset with SIFT matches. The results are shown in Table 5 (first row vs. second row). The model trained in a single stage performs similarly to the two-stage trained model in the in-scene generalization test and significantly worse in the cross-scene test.

Correspondences pruning. Previous works [58, 28, 52] used correspondence pruning to reduce the effect of the outliers' distribution on the final prediction. Specifically in these schemes, matches with the lowest classification scores were removed after each block in their networks. To test the effect of iterative pruning in our model, we implemented a similar scheme, removing half of the input matches with the lowest scores after each block. We test this pruning strategy on a model trained on YFCC with SIFT matches without keypoint denoising. In contrast to results reported for previous methods, the pruning process had a slightly negative effect on our model prediction, decreasing its mAP5° cross-scene score from 65.32% to 64.52%, probably because pruning may also remove some inlier matches. This experiment suggests that our NAC blocks can handle large numbers of outliers successfully without the need for additional pruning.

Table 5: **Ablation studies**. Evaluation of our model without keypoint denoising and 2-stage training. mAP5°(%) is reported, and the best result in each column and dataset/descriptors is marked in bold. In-scene denote results on novel image pairs taken from scenes that were included in the training data and cross-scene denote results on image pairs taken from unseen scenes.

Dataset	Descriptor	2-stage training	Keypoint denoising	In-scene	Cross-scene
YFCC	SIFT	–	–	57.31	59.70
		✓	–	58.95	65.32
		✓	✓	**60.10**	**66.14**
	SuperPoint	✓	–	54.65	58.64
		✓	✓	**55.94**	**59.10**
SUN3D	SIFT	✓	–	30.73	23.02
		✓	✓	**34.15**	**25.36**
	SuperPoint	✓	–	27.19	21.67
		✓	✓	**33.97**	**24.67**

Hyper-parameter search. We tested our model with different hyper-parameters, including the number of NAC blocks (which affects the number of DeepSet layers) and the encoder dimension. Using two NAC blocks instead of three reduces the mAP5° to 53.52% and 59.75% for in-scene and

cross-scene, respectively. Setting the encoder dimension to 256 instead of 512 reduces the mAP5° to 54.86% and 63.47%. These hyper-parameters search further justify our choices.

4.6 Implementation details

Training. At first, we trained our model on the YFCC dataset with SIFT matches (25 epochs in the noise-free pretraining stage and an additional 10 epochs in the second stage). Then, to save on resources, we finetuned the noise-free pretrained YFCC/SIFT model to initialize the training of the rest of the models (SUN3D and YFCC with SuperPoint features), for another 5 epochs with the respective noise-free dataset. Lastly, we train these pretrained models for 10 more epochs using the original (noisy) data. Training was run on an NVIDIA Quadro RTX 6000/ DGX V100/ A40 GPUs, with a maximum memory usage of 5GB.

For the loss function we set $\beta_{\text{inliers}} = 1, \beta_{\text{outliers}} = 10, \alpha_{\text{mod}} = 1$ and $\alpha_{\text{ns}} = 100$. We used a threshold of 3×10^{-3} for labeling inliers and outliers. In training, we used the ADAM[23] optimizer with a batch size of 32 image pairs and a learning rate of 10^{-4}. We note that in the noise-free pretraining stage, the predictions of all three blocks are considered in the loss function, whereas in the second stage, only the prediction of the third block is considered.

Architecture details. The network consists of three consecutive NAC blocks, where we only use the output of the last block at inference time. The Set Encoders in the NAC blocks combine 12 set layers interleaved with SoftPlus activation, layer normalization, and skip connections in a ResNet-like architecture. We set the dimension of the Set Encoder to 512. The Classification and Noise Heads consist of two-layer MLPs interleaved with an activation function. We used a SoftPlus activation for the classification head and a LeakyReLU for the noise head. The classification head produces an $n \times 2$ vector. We apply a sigmoid function on the first coordinate to predict $\hat{Y}$ and use the second coordinate for the weight prediction W.

5 Conclusion

We presented NACNet, a Noise-Aware Deep Sets framework to estimate relative camera pose, given a set of putative matches extracted from two views of a scene. We demonstrated that a position denoising of inliers and noise-free pretraining enable accurate estimation of the essential matrix. Our experiments indicate that our method can handle large numbers of outliers and achieve accurate pose estimation superior to previous methods. We generally observed good cross-dataset and cross-descriptor generalization compared to existing methods, but hope to further improve on those in future work. In addition, we believe adding a block performing degeneracy test, can further help properly utilizing non-degenerate configurations of matches and consequently improve the results of the DLT block. Finally, in future work, we will seek to incorporate our work in multiview structure from motion pipelines.

Acknowledgments and Disclosure of Funding

References

[1] Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. Communications of the ACM **54**(10), 105–112 (2011)

[2] Barath, D., Matas, J.: Graph-Cut RANSAC. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6733–6741 (2018)

[3] Barath, D., Matas, J., Noskova, J.: MAGSAC: marginalizing sample consensus. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10197–10205 (2019)

[4] Barath, D., Noskova, J., Ivashechkin, M., Matas, J.: MAGSAC++, a fast, reliable and accurate robust estimator. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1304–1312 (2020)

[5] Bay, H., Tuytelaars, T., Gool, L.V.: SURF: Speeded up robust features. In: European conference on computer vision. pp. 404–417. Springer (2006)

[6] Bradski, G.: The opencv library. Dr. Dobb's Journal: Software Tools for the Professional Programmer **25**(11), 120–123 (2000)

[7] Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Transactions on Robotics **37**(6), 1874–1890 (2021)

[8] Chum, O., Matas, J., Kittler, J.: Locally optimized RANSAC. In: Joint Pattern Recognition Symposium. pp. 236–243. Springer (2003)

[9] Chum, O., Werner, T., Matas, J.: Two-view geometry estimation unaffected by a dominant plane. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 772–779. IEEE (2005)

[10] Dai, L., Liu, Y., Ma, J., Wei, L., Lai, T., Yang, C., Chen, R.: MS^{2}DG-Net: Progressive Correspondence Learning via Multiple Sparse Semantics Dynamic Graph. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8973–8982 (2022)

[11] DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperPoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)

[12] Dusmanu, M., Schönberger, J.L., Pollefeys, M.: Multi-view optimization of local feature geometry. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. pp. 670–686. Springer (2020)

[13] Edstedt, J., Bökman, G., Wadenbäck, M., Felsberg, M.: DeDoDe: Detect, Don't Describe — Describe, Don't Detect for Local Feature Matching. In: 2024 International Conference on 3D Vision (3DV). IEEE (2024)

[14] Edstedt, J., Bökman, G., Zhao, Z.: DeDoDe v2: Analyzing and Improving the DeDoDe Keypoint Detector . In: IEEE/CVF Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2024)

[15] Eichhardt, I., Barath, D.: Optimal multi-view correction of local affine frames. arXiv preprint arXiv:1905.00519 (2019)

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

[17] Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)

[18] Heinly, J., Schonberger, J.L., Dunn, E., Frahm, J.M.: Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3287–3295 (2015)

[19] Jin, Y., Mishkin, D., Mishchuk, A., Matas, J., Fua, P., Yi, K.M., Trulls, E.: Image matching across wide baselines: From paper to practice. International Journal of Computer Vision **129**(2), 517–547 (2021)

[20] Kasten, Y., Geifman, A., Galun, M., Basri, R.: Algebraic characterization of essential matrices and their averaging in multiview settings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5895–5903 (2019)

[21] Kazerouni, I.A., Fitzgerald, L., Dooly, G., Toal, D.: A survey of state-of-the-art on visual SLAM. Expert Systems with Applications **205**, 117734 (2022)

[22] Khairuddin, A.R., Talib, M.S., Haron, H.: Review on simultaneous localization and mapping (SLAM). In: 2015 IEEE international conference on control system, computing and engineering (ICCSCE). pp. 85–90. IEEE (2015)

[23] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[24] Klopschitz, M., Irschara, A., Reitmayr, G., Schmalstieg, D.: Robust incremental structure from motion. In: Proc. 3DPVT. vol. 2, pp. 1–8 (2010)

[25] Li, Z., Zhang, S., Ma, J.: U-Match: Two-view Correspondence Learning with Hierarchy-aware Local Context aggregation. In: International Joint Conference on Artificial Intelligence (IJCAI) (2023)

[26] Lindenberger, P., Sarlin, P.E., Larsson, V., Pollefeys, M.: Pixel-perfect structure-from-motion with featuremetric refinement. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5987–5997 (2021)

[27] Lindenberger, P., Sarlin, P.E., Pollefeys, M.: LightGlue: Local Feature Matching at Light Speed. In: ICCV (2023)

[28] Liu, X., Yang, J.: Progressive Neighbor Consistency Mining for Correspondence Pruning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9527–9537 (2023)

[29] Liu, Y., Liu, L., Lin, C., Dong, Z., Wang, W.: LearnableMotion Coherence for Correspondence pruning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3237–3246 (2021)

[30] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)

[31] Luanyuan, D., Du, X., Zhang, H., Tang, J.: MGNet: Learning Correspondences via Multiple Graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 3945–3953 (2024)

[32] Ma, J., Zhao, J., Jiang, J., Zhou, H., Guo, X.: Locality preserving matching. International Journal of Computer Vision **127**(5), 512–531 (2019)

[33] Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2007)

[34] Moo Yi, K., Trulls, E., Ono, Y., Lepetit, V., Salzmann, M., Fua, P.: Learning to find good correspondences. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2666–2674 (2018). https://doi.org/10.1109/cvpr.2018.00282

[35] Moran, D., Koslowsky, H., Kasten, Y., Maron, H., Galun, M., Basri, R.: Deep permutation equivariant structure from motion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5976–5986 (2021)

[36] Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)

[37] Peyré, G., Cuturi, M., et al.: Computational optimal transport: With applications to data science. Foundations and Trends® in Machine Learning **11**(5-6), 355–607 (2019)

[38] Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30**, 5099–5108 (2017)

[39] Raguram, R., Chum, O., Pollefeys, M., Matas, J., Frahm, J.M.: USAC: A universal framework for random sample consensus. IEEE transactions on pattern analysis and machine intelligence **35**(8), 2022–2038 (2012)

[40] Ranftl, R., Koltun, V.: Deep fundamental matrix estimation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 284–299 (2018)

[41] Riba, E., Mishkin, D., Ponsa, D., Rublee, E., Bradski, G.: Kornia: an open source differentiable computer vision library for pytorch. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3674–3683 (2020)

[42] Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperGlue: Learning Feature Matching With Graph Neural Networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)

[43] Schönberger, J.L., Frahm, J.M.: Structure-from-Motion Revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

[44] Shalev-Shwartz, S., Ben-David, S.: Understanding machine learning: From theory to algorithms. Cambridge university press (2014)

[45] Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. International journal of computer vision **80**, 189–210 (2008)

[46] Sun, W., Jiang, W., Trulls, E., Tagliasacchi, A., Yi, K.M.: ACNe: Attentive Context Normalization for Robust Permutation-Equivariant Learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11286–11295 (2020)

[47] Taketomi, T., Uchiyama, H., Ikeda, S.: Visual SLAM algorithms: A survey from 2010 to 2016. IPSJ Transactions on Computer Vision and Applications **9**(1), 1–11 (2017)

[48] Tang, J., Kim, H., Guizilini, V., Pillai, S., Ambrus, R.: Neural outlier rejection for self-supervised keypoint learning. arXiv preprint arXiv:1912.10615 (2019)

[49] Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: YFCC100M: The new data in multimedia research. Communications of the ACM **59**(2), 64–73 (2016)

[50] Wagstaff, E., Fuchs, F.B., Engelcke, M., Osborne, M.A., Posner, I.: Universal approximation of functions on sets. The Journal of Machine Learning Research **23**(1), 6762–6817 (2022)

[51] Wilson, K., Snavely, N.: Robust global translations with 1dsfm. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III 13. pp. 61–75. Springer (2014)

[52] Xiangyang Miao, Guobao Xiao*, S.W., Yu, J.: BCLNet: Bilateral Consensus Learning for Two-View Correspondence Pruning. In: Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI). pp. 4225–4232 (2024)

[53] Xiao, J., Owens, A., Torralba, A.: Sun3d: A database of big spaces reconstructed using sfm and object labels. In: Proceedings of the IEEE international conference on computer vision. pp. 1625–1632 (2013)

[54] Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: Learned invariant feature transform. In: European conference on computer vision. pp. 467–483. Springer (2016)

[55] Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R.R., Smola, A.J.: Deep Sets. Advances in neural information processing systems **30** (2017)

[56] Zhang, J., Sun, D., Luo, Z., Yao, A., Zhou, L., Shen, T., Chen, Y., Quan, L., Liao, H.: Learning two-view correspondences and geometry using order-aware network. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5845–5854 (2019)

[57] Zhang, S., Ma, J.: Convmatch: Rethinking network design for two-view correspondence learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)

[58] Zhao, C., Ge, Y., Zhu, F., Zhao, R., Li, H., Salzmann, M.: Progressive Correspondence Pruning by Consensus Learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6464–6473 (2021)

A Appendix

Below, we include further quantitative evaluations, a comparison of visual results between our method and previous methods, and a list of used assets' licenses.

A.1 Denoising evaluation

Below we evaluate the utility of the denoising component of our network. Figure 5 shows the noise distribution before and after our keypoint denoising, measured according to the ground truth essential matrix. The median of the mean reprojection error (over each pair) reduces due to this component by 0.202 pixels and even more (0.246 pixels) for image pairs with pose error (maximum between the translation and rotation errors) lower than five degrees.

Figure 5: **Denoising evaluation.** Reprojection error of inlier keypoints before and after applying our denoising scheme, computed using the ground truth pose. The box plots show the 0.25, 0.5, and 0.75 quantiles. The two left bar plots represent the evaluation over all the image pairs in the YFCC dataset. The right two bar plots focus on image pairs whose pose prediction was accurate (i.e., pose error below $5°$, where the pose error is defined as the maximum of the translation and rotation angular errors.). Evaluation was conducted on the YFCC dataset using SIFT descriptors.
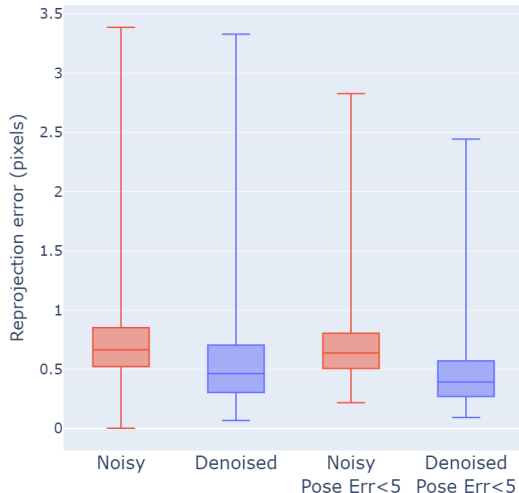
A.2 Classification evaluation

Our inlier/outlier classification results are shown in Table 6. "Ground truth" labels are determined by triangulation, as discussed in Section 3.2. It can be seen that our model achieves higher F1 scores compared to previous methods, possibly explaining our overall improved regression accuracy.

A.3 Qualitative results

Figure 6 shows results obtained with our method, compared with CLNet and MGNet. Here we use the same indoor and outdoor image pairs shown in the MGNet paper. In Figure 7 we compare our results to NCMNet (using their published checkpoint) on randomly selected image pairs from the YFCC dataset. It can be seen in both figures that our method generally outputs fewer outliers than previous methods.

A.4 License of used assets

- YFCC100M dataset[49] images are under a common-creative license, and each media file in the dataset is subject to the Creative Commons licenses chosen by their creators/uploaders.
- SUN3D dataset[53] is published under MIT license.

14

Table 6: **Classification evaluation.** Inlier/outlier classification results on the YFCC dataset and SIFT descriptors. Precision(P), Recall(R), and F1 score are reported.

Methods	In-scene			Cross-scene		
	$P(\%)$	$R(\%)$	$F1(\%)$	$P(\%)$	$R(\%)$	$F1(\%)$
RANSAC	47.4	52.6	49.9	43.5	50.6	46.8
PointNet++	49.8	86.4	63.2	46.6	84.1	59.9
LFGC	56.6	86.3	68.3	54.6	84.7	66.4
OANet++	60.0	89.3	71.8	55.7	85.9	67.6
MSA-Net	61.9	90.5	73.5	58.7	87.9	70.4
CLNet	76.0	79.2	77.6	75.0	76.4	75.7
MS^2DG-Net	63.1	90.9	74.5	59.1	88.4	70.8
ConvMatch	63.0	**91.5**	74.6	58.7	**89.3**	70.9
NCMNet	78.4	81.7	79.6	77.0	78.2	77.4
BCLNet	78.4	82.5	80.1	77.3	79.7	78.3
Ours	**84.6**	82.9	**83.2**	**82.2**	79.1	**80.2**

- Phototourism dataset[19] is published under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License: CC BY-NC-SA 4.0

- SuperGlue[42] code and weights are published under a license for:"ACADEMIC OR NON-PROFIT ORGANIZATION NONCOMMERCIAL RESEARCH USE ONLY"

- SuperPoint[11] code and weights are published under a license for:"ACADEMIC OR NON-PROFIT ORGANIZATION NONCOMMERCIAL RESEARCH USE ONLY"

- part of our code is adopted from CLNet[58] which is published under GPL-3.0 license.

- BCLNet[52] and NCMNet[28] code is published without specifying a license.

- OPENCV[6] and Kornia[41] are published under Apache License 2.0
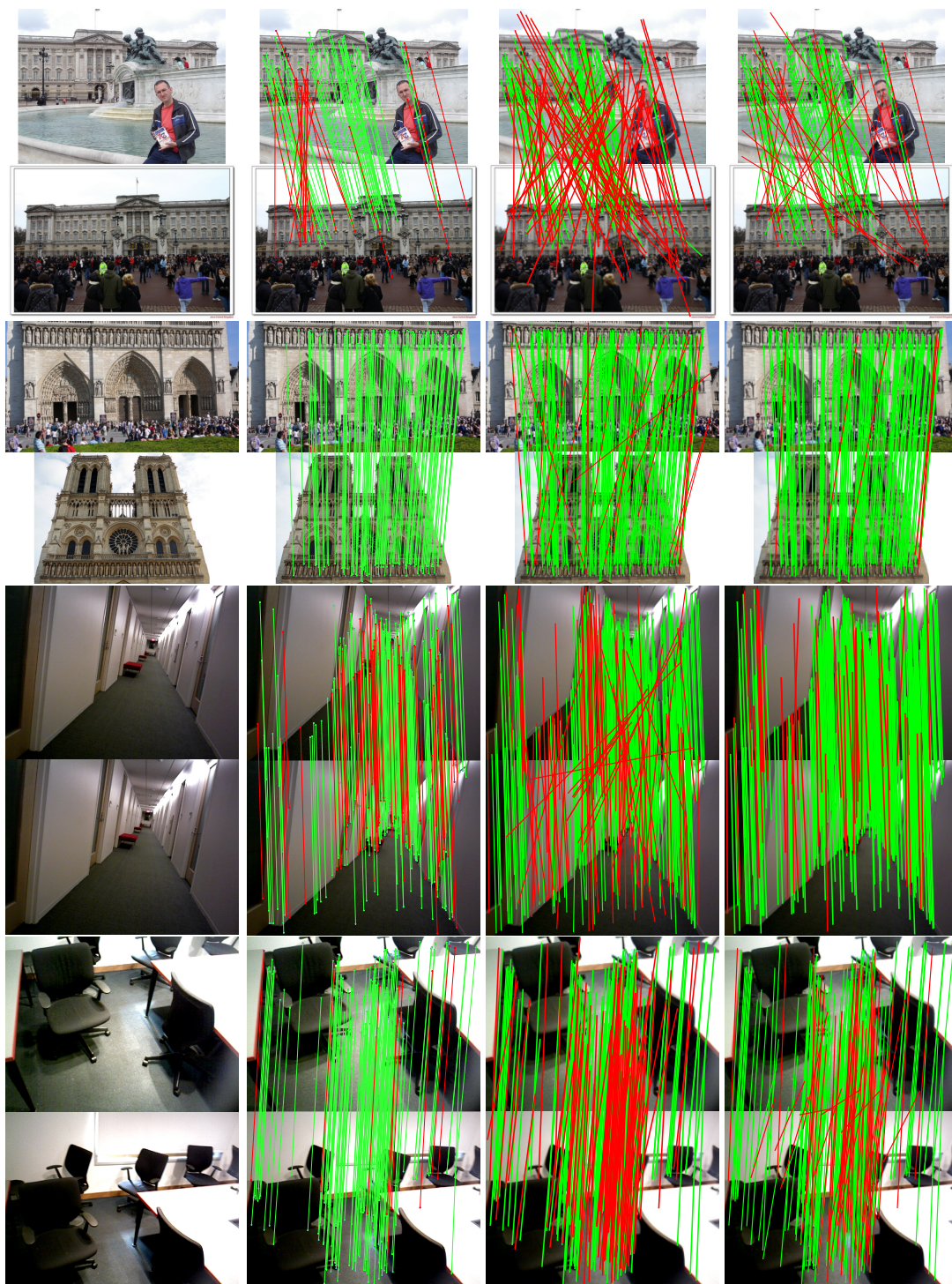
(a) Input   (b) NACNet(Ours)   (c) CLNet   (d) MGNet

Figure 6: **Qualitative results**. Visualization results of two-view correspondence pruning on unknown outdoor and indoor scenes. From left to right are the input pairs and the results of NACNet, CLNet, and MGNet, respectively. Inliers are marked in green and outliers are marked in red.

(a) Input   (b) NACNet(Ours)   (c) NCMNet

Figure 7: **Qualitative results**. Visualization results of two-view correspondence pruning on unknown outdoor scenes. From left to right are the input pairs and the results of NACNet and NCMNet, respectively. Inliers are marked in green and outliers are marked in red.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All of the claims made in the abstract and introduction are described in the method and experiment parts.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss our model limitations in the conclusion section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: There are no theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper describes our method's implementation details and training scheme in detail.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our code and data are publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All of our training and test details are written in the experiments section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The common practice in the outlier removal papers is not to report error bars and statistical significance of the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: In the implementation details section, we indicate the GPU types and memory consumption used in our training.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: We have read the code of ethics, and our research and paper are in line with it.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: Our work does not have any direct societal impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: There is no risk for misuse of our model.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All of the models and data are cited correctly, their license terms of use are respected, and explicitly mentioned in the supplementary.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We have yet to release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: We didn't use crowdsourcing or human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: We didn't use crowdsourcing or human subject.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.