RLSAC: Reinforcement Learning enhanced Sample Consensus for End-to-End Robust Estimation (Supplementary Material)

Chang Nie[1], Guangming Wang[1,3], Zhe Liu[2*], Luca Cavalli[3], Marc Pollefeys[3,4], Hesheng Wang[1*]
[1]Department of Automation, Key Laboratory of System Control
and Information Processing of Ministry of Education, Shanghai Jiao Tong University
[2] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[3] Department of Computer Science, ETH Zürich [4] Microsoft Mixed Reality and AI Zürich Lab

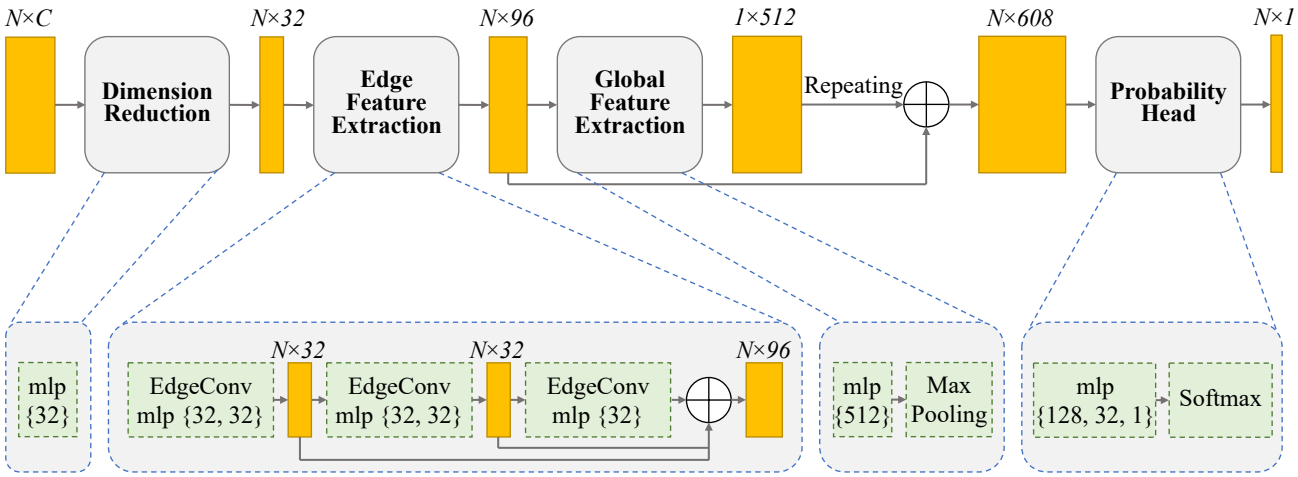


Figure 1. **The architecture of the Policy Network.**

1. Overview

In this supplementary material, we present additional results and details of our study. In Section 2, we provide the results and parameters of the policy network. Section 3 includes visualizations for both the 2D line fitting task and the fundamental matrix estimation task. In Section 4, we present ablation study for the 2D line fitting task and the network structure on the fundamental matrix estimation task.

2. Policy Network

RLSAC is built on SAC-Discrete framework [2], utilizing the policy network of actor and critic networks shown in Figure 1. The critic network does not have a final softmax operation. The architecture comprises four main components: Dimension Reduction, Edge Feature Extraction, Global Feature Extraction, and Probability Head. Since the channel of state feature is $c = 261$, we first implemented a dimension reduction module using multi-layer perceptron (MLP) to avoid information loss caused by large channel differences. Then the edge convolution (EdgeConv) [4] is used to form the Edge Feature Extraction module, which can extract edge feature effectively. We further extracted global features using MLP with Max Pooling, which are then combined with each edge feature. Finally, to output the probability of each data point belonging to the minimum set, Probability Head module uses MLPs and softmax to generate the output. The detailed network parameters are shown in Table 1. For MLP, 1×1 convolution with 1 stride is used.

3. Visualization

To better demonstrate the performance of RLSAC on both the 2D line fitting task and the fundamental matrix estimation task, we visualize RLSAC on various scenes. For the 2D line fitting task, we adopt the mean Average Accuracy (mAA) metric in [1]. The angular difference between the estimated

Table 1. **Detailed network parameters in policy network.** K Nearest Neighbors (KNN) are selected in the EdgeConv. MLP width means the number of output channels for each layer of MLP.

Module	Layer		Parameter
Dimension Reduction		MLP	*width*={*32*}
		BatchNorm	-
		LeakyReLU	*negative slope=0.2*
Edge Feature Extraction	EgdeConv 0	Get Graph Feature	*k=15*
		MLP	*width*={*32*}
		BatchNorm	-
		LeakyReLU	*negative slope=0.2*
		MLP	*width*={*32*}
		BatchNorm	-
		LeakyReLU	*negative slope=0.2*
		Max Pool	-
	EgdeConv 1	Get Graph Feature	*k=15*
		MLP	*width*={*32*}
		BatchNorm	-
		LeakyReLU	*negative slope=0.2*
		MLP	*width*={*32*}
		BatchNorm	-
		LeakyReLU	*negative slope=0.2*
		Max Pool	-
	EgdeConv 2	Get Graph Feature	*k=15*
		MLP	*width*={*32*}
		BatchNorm	-
		LeakyReLU	*negative slope=0.2*
		Max Pool	-
	Concatenate 0	EgdeConv 0 + EgdeConv 1 + EgdeConv 2	-
Global Feature Extraction		MLP	*width*={*512*}
		BatchNorm	-
		LeakyReLU	*negative slope=0.2*
		Max Pool	-
Concatenate 1		Concatenate 0 + Global Feature	-
Probability Head		MLP	*width*={*128, 32*}
		BatchNorm	-
		LeakyReLU	*negative slope=0.2*
		Dropout	*dropout rate=0.5*
		MLP	*width*={*1*}
		BatchNorm	-
		LeakyReLU	*negative slope=0.2*
		Softmax	-

line and the ground truth line is used as the error metric, which is used to calculate the mAA metric with a tolerance threshold of 0.5°. For the fundamental matrix estimation task, the mAA with threshold of 10° is used as evaluation metric. It can be seen that RLSAC can choose a suitable minimum set to explore better hypotheses.

By visualizing 2D line fitting experiments at various outlier point ratios in Figure 2, we can see how RLSAC is robust to noise interference and can efficiently explore better models. However, as shown in specific steps like Step 7 in Figure 2(e), RLSAC does not always move towards the best model. This is because the agent in RLSAC outputs actions to maximize long-term return rather than finding the best model at current step. This aligns with the exploration

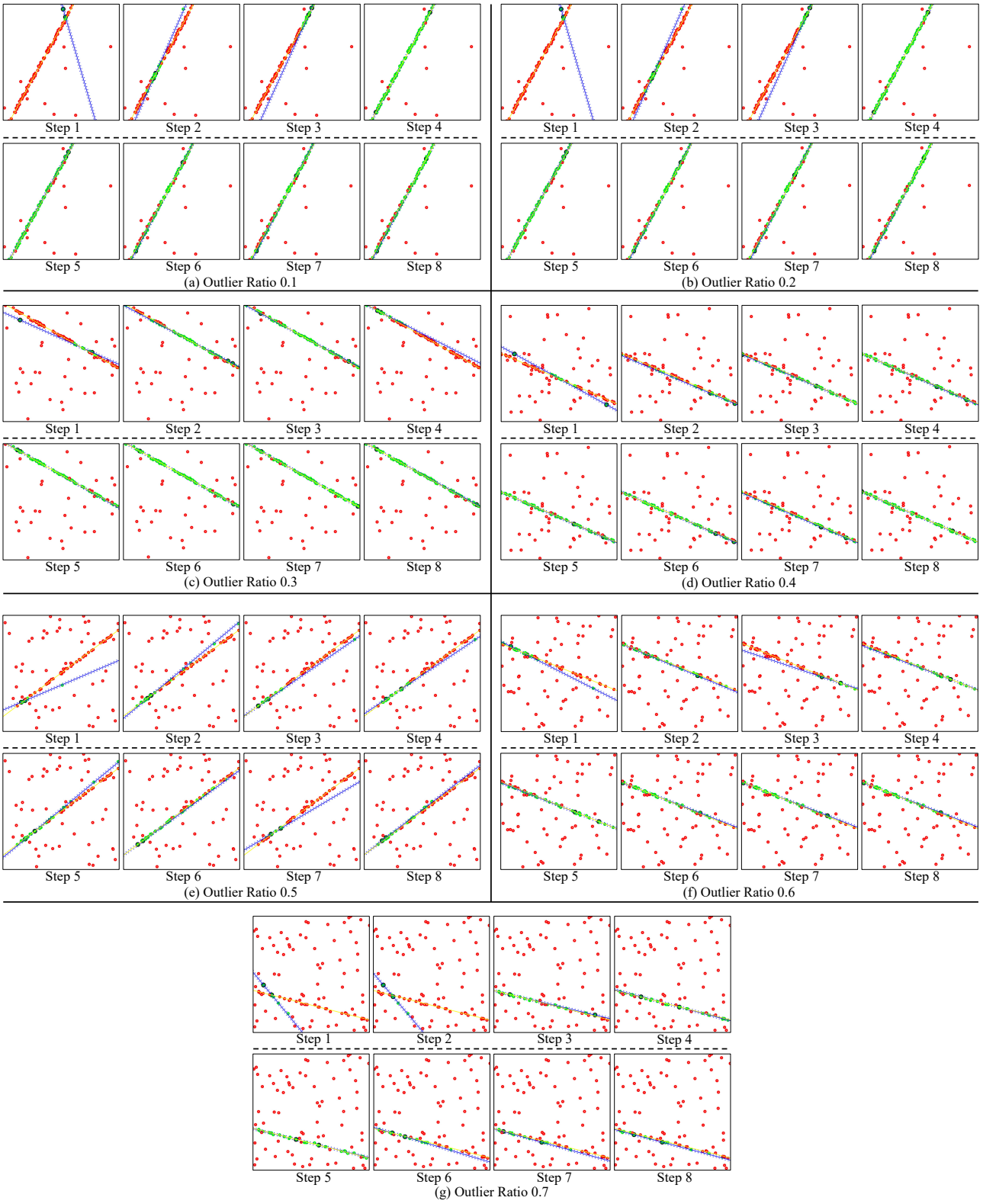
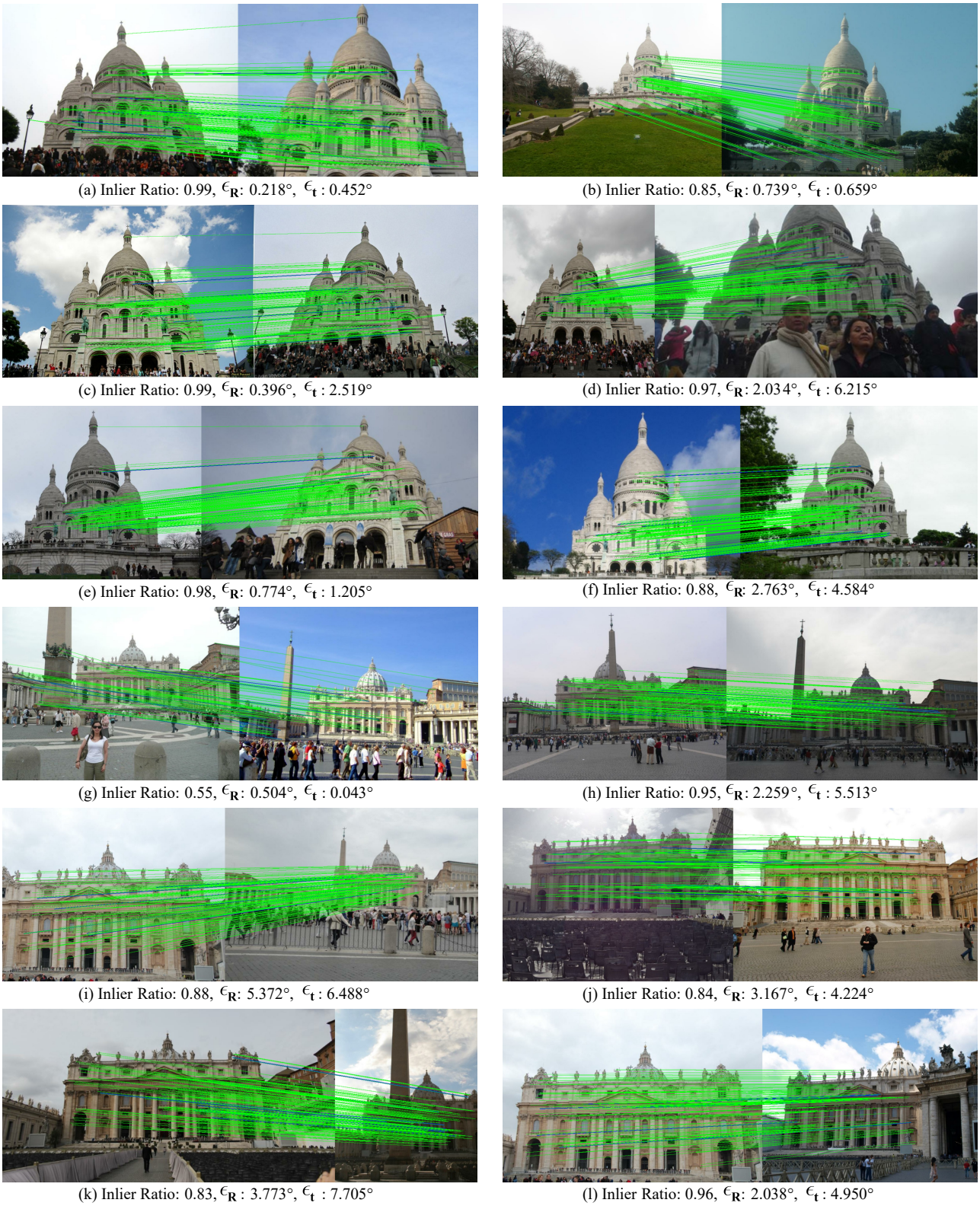


Figure 2. **The step results of RLSAC on 2D line fitting in different outlier ratio.** The green points represent inliers, while the red points represent outliers. The sampled minimum set points are denoted by black edges. The ground truth is represented by a yellow line, while the hypothesis and inlier threshold are represented by blue and dashed lines respectively.



(a) Inlier Ratio: 0.99, $\epsilon_{\mathbf{R}}$: 0.218°, $\epsilon_{\mathbf{t}}$: 0.452°

(b) Inlier Ratio: 0.85, $\epsilon_{\mathbf{R}}$: 0.739°, $\epsilon_{\mathbf{t}}$: 0.659°

(c) Inlier Ratio: 0.99, $\epsilon_{\mathbf{R}}$: 0.396°, $\epsilon_{\mathbf{t}}$: 2.519°

(d) Inlier Ratio: 0.97, $\epsilon_{\mathbf{R}}$: 2.034°, $\epsilon_{\mathbf{t}}$: 6.215°

(e) Inlier Ratio: 0.98, $\epsilon_{\mathbf{R}}$: 0.774°, $\epsilon_{\mathbf{t}}$: 1.205°

(f) Inlier Ratio: 0.88, $\epsilon_{\mathbf{R}}$: 2.763°, $\epsilon_{\mathbf{t}}$: 4.584°

(g) Inlier Ratio: 0.55, $\epsilon_{\mathbf{R}}$: 0.504°, $\epsilon_{\mathbf{t}}$: 0.043°

(h) Inlier Ratio: 0.95, $\epsilon_{\mathbf{R}}$: 2.259°, $\epsilon_{\mathbf{t}}$: 5.513°

(i) Inlier Ratio: 0.88, $\epsilon_{\mathbf{R}}$: 5.372°, $\epsilon_{\mathbf{t}}$: 6.488°

(j) Inlier Ratio: 0.84, $\epsilon_{\mathbf{R}}$: 3.167°, $\epsilon_{\mathbf{t}}$: 4.224°

(k) Inlier Ratio: 0.83, $\epsilon_{\mathbf{R}}$: 3.773°, $\epsilon_{\mathbf{t}}$: 7.705°

(l) Inlier Ratio: 0.96, $\epsilon_{\mathbf{R}}$: 2.038°, $\epsilon_{\mathbf{t}}$: 4.950°

Figure 3. **The qualitative results of RLSAC on the fundamental matrix estimation task in different scenes.** Inlier rate, rotation and translation errors are reported. The blue lines represent the sampled minimum set points, and the green lines represent the inliers.

Table 2. **Ablation study on 2D line fitting with different number of points.** The mAA@0.5° and median error($^\circ$) on various outlier ratios are reported. -200 means training in 100 points and testing in 200 points. All methods are iterated 150 times.

Method	0.1		0.2		0.3		0.4		0.5		0.6		0.7	
	mAA ↑	Mid. ↓	mAA ↑	Mid. ↓	mAA ↑	Mid. ↓	mAA ↑	Mid. ↓	mAA ↑	Mid. ↓	mAA ↑	Mid. ↓	mAA ↑	Mid. ↓
RANSAC [3]	0.870	0.049	0.863	0.052	0.850	0.056	0.829	0.061	0.796	0.071	0.746	0.087	0.608	0.135
Ours	**0.875**	**0.047**	**0.874**	**0.049**	**0.872**	**0.048**	**0.865**	**0.050**	**0.858**	**0.052**	**0.845**	**0.056**	**0.824**	**0.062**
RANSAC-200 [3]	**0.874**	**0.048**	0.867	0.050	0.855	0.054	0.830	0.062	0.797	0.073	0.723	0.097	0.627	0.130
Ours-200	0.864	0.050	**0.871**	**0.049**	**0.864**	**0.050**	**0.851**	**0.053**	**0.850**	**0.053**	**0.828**	**0.061**	**0.821**	**0.063**

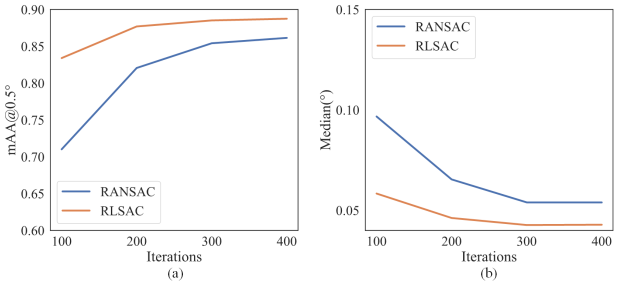


Figure 4. **The mAA@0.5° and median error($^\circ$) at different iterations.** The results on the 2D line fitting task at different iterations with 0.5 outlier ratio is reported.

Table 3. **The additional ablation study results** of RLSAC on fundamental matrix estimation. DIM represents dimension.

	Method	mAA@10° ↑		Median ($^\circ$) ↓	
		R	**t**	$\epsilon_{\mathbf{R}}$	$\epsilon_{\mathbf{t}}$
(a)	Ours (with $k = 10$ nearest neighbors)	0.754	0.618	0.975	1.907
	Ours (with $k = 20$ nearest neighbors)	0.754	0.611	0.996	1.983
	Ours (full, with $k = 15$ nearest neighbors)	**0.760**	**0.622**	**0.926**	**1.751**
(b)	Ours (with 32 DIM SVD descriptors reduction)	0.702	0.572	1.371	2.882
	Ours (with 64 DIM SVD descriptors reduction)	0.698	0.566	1.450	2.998
	Ours (with 16 DIM MLP reduction)	0.755	0.611	0.993	2.002
	Ours (with 64 DIM MLP reduction)	0.757	0.616	0.977	1.929
	Ours (full, with 32 DIM MLP reduction)	**0.760**	**0.622**	**0.926**	**1.751**
(c)	Ours (with 16 DIM EdgeConv)	0.755	0.612	1.002	1.945
	Ours (with 64 DIM EdgeConv)	0.753	0.616	0.999	1.928
	Ours (full, with 32 DIM EdgeConv)	**0.760**	**0.622**	**0.926**	**1.751**
(d)	Ours (with 256 DIM global MLP)	0.751	0.614	0.989	1.921
	Ours (with 1024 DIM global MLP)	0.754	0.613	1.003	1.918
	Ours (full, with 512 DIM global MLP)	**0.760**	**0.622**	**0.926**	**1.751**
(e)	Ours (with (64, 16) DIM head MLPs)	0.754	0.618	0.994	1.948
	Ours (with (256, 64) DIM head MLPs)	0.755	0.615	0.976	1.943
	Ours (full, with (128, 32) DIM head MLPs)	**0.760**	**0.622**	**0.926**	**1.751**

strategy in reinforcement learning, where RLSAC may move into a state with lower current reward to better understand the environment and explore optimal strategies.

The visualization of RLSAC on the fundamental matrix estimation task is presented in Figure 3. It reveals that RLSAC is effective in estimating accurate fundamental matrix to solve the precise pose transformation. Moreover, RLSAC performs well even with large variations in time, space, illumination, and dynamic objects.

4. Ablation Study

We perform additional ablation studies on 2D line fitting and fundamental matrix estimation tasks. The experimental data and settings remain the same as in Section 5 of the main text.

Robustness and Generalization of RLSAC in 2D Line Fitting: The Ours-200 from Table 2 means that RLSAC is trained using 100 points in each outlier ratio and then tested using 200 points. For comparison, RANSAC is also tested using 200 points. In addition, the 100 points results are presented as RANSAC and Ours. Ours-200 outperforms RANSAC in most scenes, while only incurring a modest reduction in performance compared to Ours. This shows the great robustness and generalization of RLSAC across varying numbers of data.

Efficiency of RLSAC in 2D Line Fitting: The Figure 4 illustrates that RLSAC requires only fewer iterations to achieve the same performance compared to RANSAC for the same outlier ratio. Moreover, as the iterations increases, the performance gains of both methods decrease. However, RLSAC can leverage memory features and data features for sampling, which enables it to sustain higher performance.

Nearest Neighbors in EdgeConv: The number of nearest neighbors k used in EdgeConv can affect the policy network to extract edge feature. In Table 3(a), different values of k are evaluated and the $k = 15$ performs best in helping RLSAC learn edge features.

Effect of Dimension Reduction: To reduce the computational cost caused by high-dimensional data features, the state feature need to be reduced. We attempted to reduce the dimension of the 128 dimensional descriptors using singular value decomposition (SVD). However, as shown in Table 3(b), this approach led to significant performance degradation due to the loss of features.

In contrast, RLSAC uses a MLP to reduce the dimension of state feature in the policy network for computational efficiency. The experimental results, as shown in Table 3(b), demonstrate that the best performance is obtained when the dimension of $c = 261$ data is reduced to $c = 32$ using the MLP. The large channel gap causing information loss during the compression process, resulting in a decline in feature extraction ability.

Different Sizes of EdgeConv: We investigate the impact of different sizes of EdgeConv on the task of fundamental matrix estimation. The experiments in Table 3(c) indicate that the 32-size EdgeConv module can extract features more effectively, while larger sizes lead to decreased performance. This observation may be attributed to the risk of overfitting with overly complex models. Furthermore, comparable performance can be attained with smaller network. This means by optimizing the network structure further, high performance can potentially be achieved using a smaller network.

Different Sizes of Global MLP: The performance of the MLP size in the global feature extraction module is evaluated and presented in Table 3(d). The results indicate that an MLP size of 512 has higher performance. This can be attributed to the fact that smaller feature vectors may struggle to encapsulate complex global features, while excessively large feature vectors may hinder subsequent information extraction.

Different Sizes of Probability Head: We conduct experiments to evaluate different sizes of head MLPs. The results in Table 3(e) show that the best performance is achieved when using head MLP with sizes (128, 32). The reason may be the same as using MLP for dimension reduction, the large channel gap may lead to information loss.

References

[1] D Barath, TJ Chin, O Chum, D Mishkin, R Ranftl, and J Matas. RANSAC in 2020 tutorial. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[2] Petros Christodoulou. Soft actor-critic for discrete action settings. *arXiv preprint arXiv:1910.07207*, 2019.

[3] 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.

[4] 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 (tog)*, 38(5):1–12, 2019.