Learning Correspondence Uncertainty via Differentiable Nonlinear Least Squares

Supplementary Material

Dominik Muhle[1] Lukas Koestler[1] Krishna Murthy Jatavallabhula[2] Daniel Cremers[1]
[1]Technical University of Munich [2]Massachusetts Institute of Technology
{dominik.muhle, lukas.koestler, cremers}@tum.de jkrishna@mit.edu

A. Overview

This supplementary material presents additional insight into learning positional uncertainty using differentiable nonlinear least squares (DNLS). We start by addressing limitations of our framework in Sec. B. Sec. C gives a derivation of the residual variance σ_s^2 for the symmetric probabilistic normal epipolar constraint (PNEC). We investigate the unobservabilities of the gradient in Sec. D. The training and evaluation details are given in Sec. E. We show further quantitative evaluations in Sec. F and Sec. G. This includes examples of how the learned covariances move the minimum around the ground truth and the results on the sequences 00-07 of the KITTI [2] dataset. We compare our learned covariances against error estimates from reprojection using ground truth poses.

B. Limitations

In this section, we will address limitations of our method, not mentioned in the main paper due to constrained space. We learn to estimate the noise distribution of keypoint detectors, using regression on the pose error. The gradient we use for learning the distribution is restricted to points in the image that are detected as keypoints. This restrict our method to learn only on regions of the image with a high chance of producing keypoints. While we don't need uncertainty information for regions without keypoints, this sparse information flow might reduce generalization capabilities to different datasets. Sparsity if further enhanced by using RANSAC to filter outliers, removing points that are too far off. However, we choose to include RANSAC for our training to obtain better pose estimates for gradients not dominated by outliers. We tried to mitigate the effect of overfitting on keypoint positions by cropping the images, leading to different keypoint positions. Furthermore, our experiments showed that generalization between KITTI and EuRoC are possible.

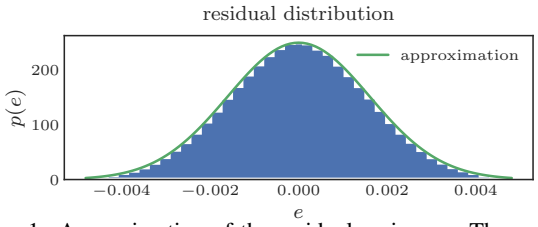


Figure 1. Approximation of the residual variances. The analytical approximation given in the main paper accurately models the true distribution of the residual given a similar setup to the KITTI dataset.

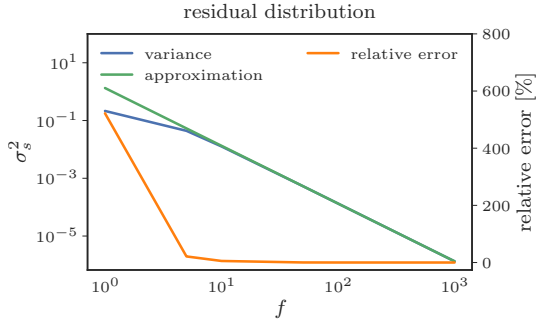
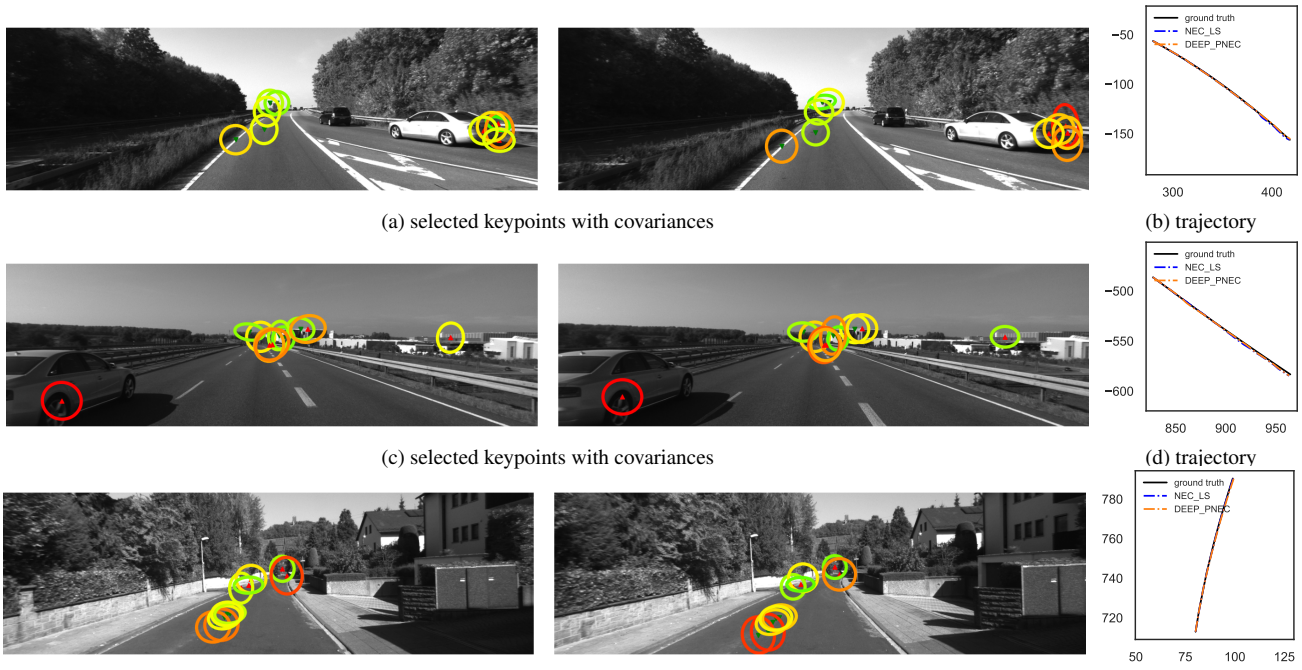


Figure 2. Approximation of the residual variances over different focal lengths. The scale of the variance is correlated with the focal length. Our approximation is better, the smaller the variance is. For a focal length similar to the one found in the KITTI dataset, the relative error is 0.015%.

Fig. 3 and Fig. 4 show examples where our method performs worse and better than the NEC-LS optimization based on the estimated covariances. We investigate the keypoints with the highest and lowest reprojection error. As Fig. 3 shows, our method is not always able to compensate keypoints on dynamic objects leading to a large rotational error. The trajectories in Fig. 4 show the improvements our method is able to achieve compared to NEC-LS.

C. Approximating σ_s^2

This section show derives the residual variance from the bearing vector covariances in both images. Given both bear-



(a) selected keypoints with covariances (b) trajectory

(c) selected keypoints with covariances (d) trajectory

(e) selected keypoints with covariances (f) trajectory

Figure 3. Left: estimated keypoints with covariances (color-coded ellipses) for examples where our method performs worse than NEC-LS. Good (▼) and bad correspondences (▲) based on the reprojection error. Right: corresponding sections of the trajectory. (a) and (c) show examples with keypoints on dynamic objects. Although their estimated covariances is somewhat lower (especially in (c)) this is not enough to compensate the error. (e) shows an example where points with a higher reprojection error get assigned a covariances on a similar level or slightly better than good correspondences.

ing vectors $\boldsymbol{f}$ and $\boldsymbol{f'}$ are noisy, we can write them as

$$\boldsymbol{f} = \boldsymbol{\mu} + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \tag{1}$$
$$\boldsymbol{f} = \boldsymbol{\mu}' + \boldsymbol{\eta}', \quad \boldsymbol{\eta}' \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}'), \tag{2}$$

with a constant and a noise term. We then get the new normal vector as

$$\begin{aligned} \boldsymbol{n}_s &= (\boldsymbol{\mu} + \boldsymbol{\eta}) \times \boldsymbol{R}(\boldsymbol{\mu}' + \boldsymbol{\eta}') \\ &= \hat{\boldsymbol{\mu}}\boldsymbol{R}\boldsymbol{\mu}' + \hat{\boldsymbol{\mu}}\boldsymbol{R}\boldsymbol{\eta}' + \hat{\boldsymbol{\eta}}\boldsymbol{R}\boldsymbol{\mu}' + \hat{\boldsymbol{\eta}}\boldsymbol{R}\boldsymbol{\eta}', \end{aligned} \tag{3}$$

with a constant term $\boldsymbol{\mu}_n = \hat{\boldsymbol{\mu}}\boldsymbol{R}\boldsymbol{\mu}'$ and a noise term $\boldsymbol{\eta}_n = \boldsymbol{R}\boldsymbol{\eta}' + \hat{\boldsymbol{\eta}}\boldsymbol{R}\boldsymbol{\mu}' + \hat{\boldsymbol{\eta}}\boldsymbol{R}\boldsymbol{\eta}'$. The noise term is zero centered and has a variance of

$$\boldsymbol{\Sigma}_n = (\hat{\boldsymbol{R}\boldsymbol{\mu}'_i})\boldsymbol{\Sigma}_i(\hat{\boldsymbol{R}\boldsymbol{\mu}'_i})^\top + \hat{\boldsymbol{\mu}}_i\boldsymbol{R}\boldsymbol{\Sigma}'_i\boldsymbol{R}^\top\hat{\boldsymbol{\mu}}_i{}^\top + \tilde{\boldsymbol{\Sigma}}, \tag{4}$$

where $\tilde{\boldsymbol{\Sigma}}$ is constructed from the columns of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}'_R = \boldsymbol{R}\boldsymbol{\Sigma}'\boldsymbol{R}^\top$ as

$$\tilde{\boldsymbol{\Sigma}} = \begin{bmatrix} (\boldsymbol{\Sigma}_2 \times \boldsymbol{\Sigma}'_{R,3} + \boldsymbol{\Sigma}_3 \times \boldsymbol{\Sigma}'_{R,2})^\top \\ (\boldsymbol{\Sigma}_3 \times \boldsymbol{\Sigma}'_{R,1} + \boldsymbol{\Sigma}_1 \times \boldsymbol{\Sigma}'_{R,3})^\top \\ (\boldsymbol{\Sigma}_1 \times \boldsymbol{\Sigma}'_{R,2} + \boldsymbol{\Sigma}_2 \times \boldsymbol{\Sigma}'_{R,1})^\top \end{bmatrix}. \tag{5}$$

As stated in the main paper, we use an approximation of the noise distribution. Since $\tilde{\boldsymbol{\Sigma}}$ is order of magnitudes smaller than the other terms, we can approximate $\boldsymbol{\Sigma}_n$ as

$$\boldsymbol{\Sigma}_n \approx (\hat{\boldsymbol{R}\boldsymbol{\mu}'_i})\boldsymbol{\Sigma}_i(\hat{\boldsymbol{R}\boldsymbol{\mu}'_i})^\top + \hat{\boldsymbol{\mu}}_i\boldsymbol{R}\boldsymbol{\Sigma}'_i\boldsymbol{R}^\top\hat{\boldsymbol{\mu}}_i{}^\top. \tag{6}$$

The final residual variance is given by

$$\sigma_s^2 = \boldsymbol{t}^\top\boldsymbol{\Sigma}_n\boldsymbol{t}. \tag{7}$$

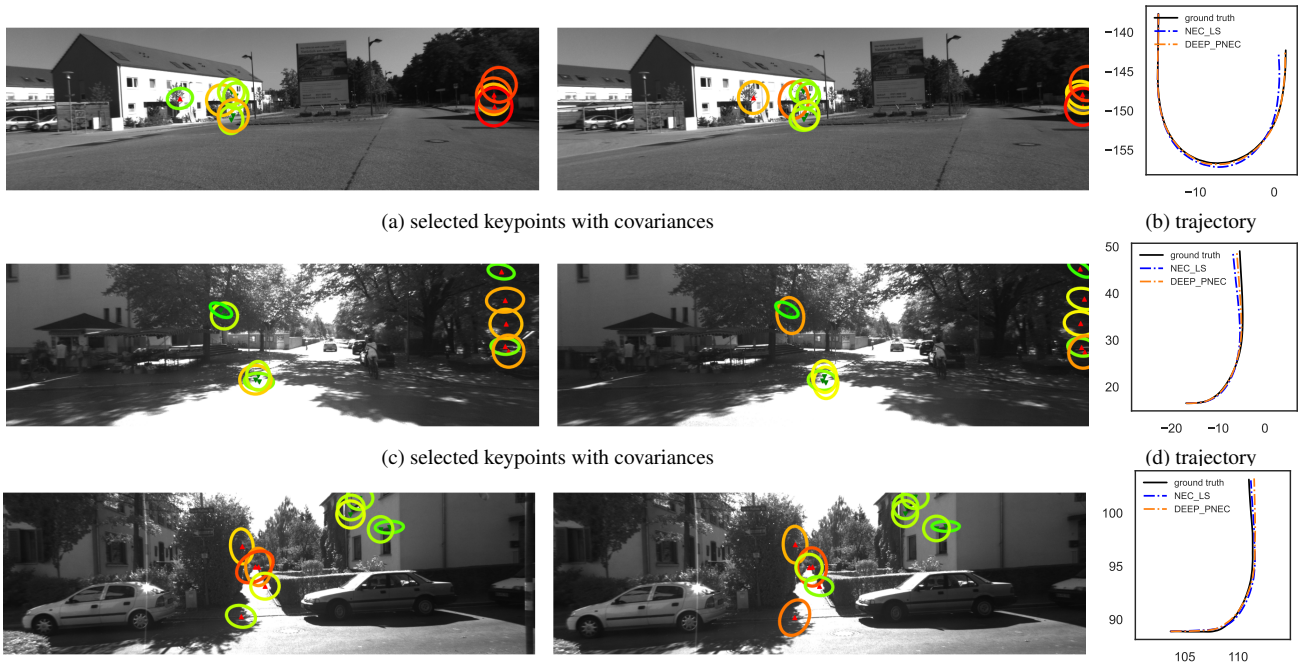
Fig. 1 shows a comparison between our approximation and a the true residual distribution, given noisy image points. Do to the unprojection of the image points to bearing vectors, the trace of the bearing vector covariances is small for a focal length f of ca. 720 pixels on the KITTI dataset, since $\mathrm{tr}(\boldsymbol{\Sigma}) \sim 1/f^2$. Given the small covariances, $\tilde{\boldsymbol{\Sigma}}$ is several magnitudes smaller than the other terms, making the approximation accurate. Fig. 2 shows the correlation between the variance and the focal length.

D. Gradient

In this section, we show that the gradient $\partial\mathcal{L}/\partial\boldsymbol{\Sigma}_{2\mathrm{D}}$ is restricted by the problem geometry. We state the components needed to obtain $\partial\mathcal{L}/\partial\boldsymbol{\Sigma}_{2\mathrm{D}}$ and show, how the geometry restricts their direction. Therefore, given a constant geometry the overall gradient direction only moves little throughout the training.

We start by rewriting the residual e_s of symmetric PNEC energy function as

$$e_s = \frac{n}{\sigma_s} = \frac{n}{\sqrt{d_{\boldsymbol{\Sigma}} + d_{\boldsymbol{\Sigma}'}}}, \tag{8}$$



(a) selected keypoints with covariances (b) trajectory

(c) selected keypoints with covariances (d) trajectory

(e) selected keypoints with covariances (f) trajectory

Figure 4. Left: estimated keypoints with covariances (color-coded ellipses) for examples where our method performs better than NEC-LS. Good (▼) and bad correspondences (▲) based on the reprojection error. Right: corresponding sections of the trajectory. Covariances for bad correspondences are estimated to be higher in these examples. They are down-weighted in the optimization leading to better pose estimates.

for easier differentiation, with the components

$$n = \boldsymbol{t}^\top \hat{\boldsymbol{f}} \exp \hat{\boldsymbol{x}} \boldsymbol{R} \boldsymbol{f}' \boldsymbol{f}'^\top \boldsymbol{R}^\top \exp \hat{\boldsymbol{x}}^\top \hat{\boldsymbol{f}}^\top \boldsymbol{t}\,, \tag{9}$$

$$d_{\boldsymbol{\Sigma}} = \left(\left(\exp \hat{\boldsymbol{x}} \boldsymbol{R} \boldsymbol{f}'\right) \times \boldsymbol{t}\right)^\top \boldsymbol{\Sigma} \left(\left(\exp \hat{\boldsymbol{x}} \boldsymbol{R} \boldsymbol{f}'\right) \times \boldsymbol{t}\right)\,, \tag{10}$$

$$d_{\boldsymbol{\Sigma}'} = \boldsymbol{t}^\top \hat{\boldsymbol{f}} \exp \hat{\boldsymbol{x}} \boldsymbol{R} \boldsymbol{\Sigma}' \boldsymbol{R}^\top \exp \hat{\boldsymbol{x}}^\top \hat{\boldsymbol{f}}^\top \boldsymbol{t}\,. \tag{11}$$

Since we are working with rotations in $\mathrm{SO}(3)$ we differentiate with regard to $\boldsymbol{x} \in \mathrm{so}(3)$ around the identity rotation. This gives us the following gradients

$$\frac{\partial n}{\partial \boldsymbol{x}} = 2 \left(\left(\boldsymbol{R} \boldsymbol{f}' \boldsymbol{f}'^\top \boldsymbol{R}^\top \exp \hat{\boldsymbol{x}}^\top \hat{\boldsymbol{f}}^\top \boldsymbol{t} \right) \times \left(\hat{\boldsymbol{f}} \boldsymbol{t} \right) \right)^\top\,, \tag{12}$$

$$\frac{\partial d_{\boldsymbol{\Sigma}}}{\partial \boldsymbol{x}} = 2 \left(\left(\boldsymbol{R} \boldsymbol{f}' \right) \times \left(\hat{\boldsymbol{t}} \boldsymbol{\Sigma} \hat{\boldsymbol{t}}^\top \exp \hat{\boldsymbol{x}} \boldsymbol{R} \boldsymbol{f}' \right) \right)^\top\,, \tag{13}$$

$$\frac{\partial d_{\boldsymbol{\Sigma}'}}{\partial \boldsymbol{x}} = 2 \left(\left(\boldsymbol{R} \boldsymbol{\Sigma}' \boldsymbol{R}^\top \exp \hat{\boldsymbol{x}}^\top \hat{\boldsymbol{f}}^\top \boldsymbol{t} \right) \times \left(\hat{\boldsymbol{f}} \boldsymbol{t} \right) \right)^\top\,, \tag{14}$$

with regard to the rotation. The direction of each gradient is restricted by the cross product. The gradient for the residual is given by

$$\frac{\partial e_s}{\partial \boldsymbol{x}} = \frac{1}{\sigma_s} \frac{\partial n}{\partial \boldsymbol{x}} - \frac{n}{2\sigma_s^3} \left(\frac{\partial d_{\boldsymbol{\Sigma}}}{\partial \boldsymbol{x}} + \frac{\partial d_{\boldsymbol{\Sigma}'}}{\partial \boldsymbol{x}} \right)\,. \tag{15}$$

The gradients with regard to the bearing vector covariances are solely dependent on the geometry as they are given by

$$\frac{\partial d_{\boldsymbol{\Sigma}}}{\partial \boldsymbol{\Sigma}} = \left(\boldsymbol{t} \times \left(\exp \hat{\boldsymbol{x}} \boldsymbol{R} \boldsymbol{f}' \right) \right) \left(\boldsymbol{t} \times \left(\exp \hat{\boldsymbol{x}} \boldsymbol{R} \boldsymbol{f}' \right) \right)^\top\,, \tag{16}$$

$$\frac{\partial d_{\boldsymbol{\Sigma}'}}{\partial \boldsymbol{\Sigma}'} = \left(\boldsymbol{R}^\top \exp \hat{\boldsymbol{x}}^\top \hat{\boldsymbol{f}}^\top \boldsymbol{t} \right) \left(\boldsymbol{R}^\top \exp \hat{\boldsymbol{x}}^\top \hat{\boldsymbol{f}}^\top \boldsymbol{t} \right)^\top\,. \tag{17}$$

The gradients of the residual are given by

$$\frac{\partial e_s}{\partial \boldsymbol{\Sigma}} = -\frac{n}{2\sigma_s^3} \frac{\partial d_{\boldsymbol{\Sigma}}}{\partial \boldsymbol{\Sigma}}\,, \tag{18}$$

$$\frac{\partial e_s}{\partial \boldsymbol{\Sigma}'} = -\frac{n}{2\sigma_s^3} \frac{\partial d_{\boldsymbol{\Sigma}'}}{\partial \boldsymbol{\Sigma}'}\,. \tag{19}$$

Since all components are restricted by the geometry of the problem, the overall gradient is somewhat restricted as well. We show this empirically in the following.

Fig. 5 and Fig. 6 give the distribution of the gradient for the first experiment on synthetic data, where all individual problems share the same geometric setup. Fig. 6 shows the eigenvectors of $\partial \mathcal{L} / \partial \boldsymbol{\Sigma}_{2\mathrm{D}}$ for one covariance in the image plane. After 10 epochs of training, the eigenvectors are mainly located at 4 distinct regions, showing the restriction of the gradient direction. Even after 100 epochs of training certain regions show only few eigenvectors. The angular distribution of the eigenvectors in Fig. 5 show 4 distinct peaks, with almost no eigenvectors in between.

Fig. 7 and Fig. 8 show the distribution of the gradient for the second experiment on synthetic data, with more diverse

Hyperparameter	KITTI	EuRoC
optimizer	ADAM	ADAM
β_1	0.9	0.9
β_2	0.999	0.999
learning rate	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$
PNEC and theseus		
regularization	10^{-13}	10^{-13}
damping	10^{7}	10^{7}
iterations	100	100
RANSAC		
iterations	5000	5000
threshold	10^{-6}	$8 \cdot 10^{-7}$

Table 1. Parameters used for training and evaluation.

data. Given the diverse data, there are eigenvectors in all directions, even after 10 epochs. Fig. 7 still shows 4 distinct peaks, however there is no sparsity in the distribution.

The sparse distribution of the gradient direction prohibit learning the correct noise distribution for the first experiment. Only the residual variance is correctly estimated. However, the introduction of diverse data with different geometries removes this restriction, leading better covariance estimates.

E. Hyperparameters

This section details the training and evaluation parameters for our DNLS framework for estimating noise distributions of keypoints. All models are trained on two RTX 5000 GPUs with 16GB of memory for around 3 days. We use a UNet architecture with 3 output channels for predicting the uncertainty parameters. The UNet has 4 down convolutions and 4 up convolutions with $32, 64, 128, 256$ and $128, 64, 32, 16$ channels, respectively. Tab. 2 gives the SuperPoint and SuperGlue hyperparameters for training and evaluation. For our supervised training, we train on consecutive image pairs of the training sequences. For our self-supervised training we create the training tuples from 3 consecutive images. When training with SuperPoint, we crop the images to size (1200, 300), whereas for KLT-Tracks, we crop it to (1200, 320). We found that reducing the height too much for KLT-tracks leads to not enough trackAss. For evaluating with KLT-tracks on KITTI we change the following to [6]: instead of tracking keypoints over multiple images, we start with fresh keypoints for each image pair. To account for the symmetric PNEC, we slightly modify the uncertainty extraction. We use [6, suppl., Eqn. (8)] as the uncertainty measure for the tracks in both frames. We found, that these changes already give better results than the ones stated [6]. Tab. 1 gives the training parameter for optimizer, theseus and the PNEC energy function not stated in the main paper.

Hyperparameter	training	KITTI	EuRoC
max keypoints	256	2048	1024
keypoint threshold	0.005	0.005	0.0005
nms radius	3	3	3
weights	outdoor	outdoor	indoor
sinkhorn iterations	20	20	20
match threshold	0.5	0.5	0.01

Table 2. Hyperparameters for SuperPoint and SuperGlue during training and evaluation on the KITTI and EuRoC dataset.

F. Moving the Minimum

Fig. 10 and Fig. 9 show examples for energy functions around the ground truth pose on the KITTI dataset. The energy functions are evaluated with keypoints filtered using the reprojection error also used in the RANSAC scheme of [3] to remove outliers. We show the energy functions evaluated for rotations around the ground truth for yaw and pitch. While the overall shape of the energy function stays the same, our methods moves the minimum closer to the ground truth pose by learning the covariances.

G. Further Results

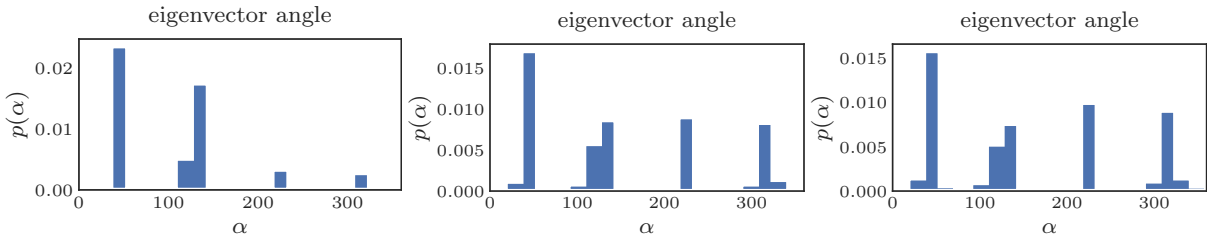
In this section we present additional results on the KITTI dataset, not presented in the main paper due to constrained space. We give the evaluation results for all sequences, training and test set. To present more comparisons with baseline methods, we replace the Nistér-5pt [8] with the 8pt [5] algorithm. Furthermore, we replace the weighted NEC-LS and the KLT-PNEC. Instead, we add another PNEC method, where we approximate the error distribution using a reprojection error. Following [3], we triangulate a 3D point using the feature correspondence $\boldsymbol{p}_i, \boldsymbol{p}'_i$ and the ground truth pose. We reproject the point into the images as $\tilde{\boldsymbol{p}}_i, \tilde{\boldsymbol{p}}'_i$ and approximate the the error distribution as scaled isotropic covariances

$$\boldsymbol{\Sigma}_{2\mathrm{D},i} = \|\tilde{\boldsymbol{p}}_i - \boldsymbol{p}_i\|^2 \boldsymbol{I}_2 \,, \tag{20}$$

$$\boldsymbol{\Sigma}'_{2\mathrm{D},i} = \|\tilde{\boldsymbol{p}}'_i - \boldsymbol{p}'_i\|^2 \boldsymbol{I}_2 \,. \tag{21}$$

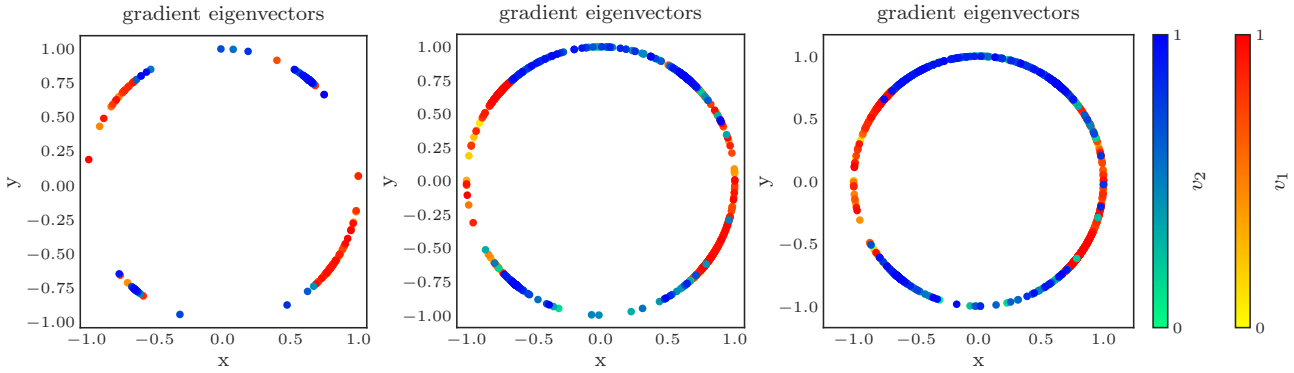
We clip the scale of the covariances at 0.01 and 4.0. Tab. 4 shows the results for the training and test set on KITTI with SuperPoint. While the reprojection method achieves the best results for the RPE_1 and e_t, our methods are often not far behind. This shows, that our network is capable and not too far off, when it comes to pose estimation. Tab. 3 shows the results for KITTI with KLT-tracks.

We show trajectories for all sequences of the KITTI dataset in Fig. 12 and Fig. 11. Our method consistently achieves the smallest drift over all sequences.



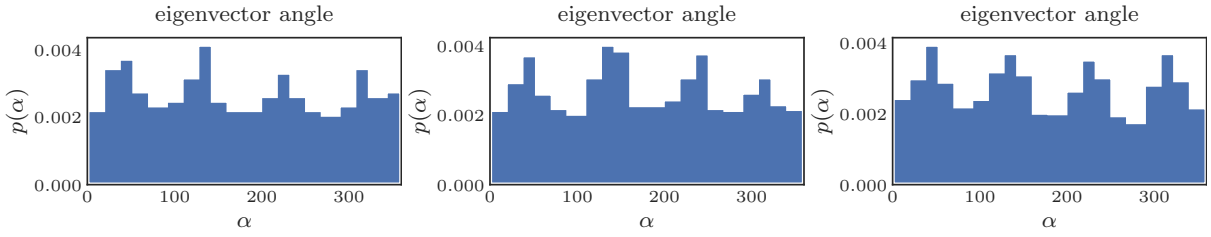
(a) 10 epochs (b) 50 epochs (c) 100 epochs

Figure 5. Histogram of eigenvector angles for the gradient $\partial\mathcal{L}/\partial\mathbf{\Sigma}'_{2D}$ after 10, 50, and 100 epochs. The histogram shows 4 distinct peaks, with only a few points in between. This shows the limited direction that the gradients have, making it difficult to learn the true distribution of the covariances with little diversity in the training data.



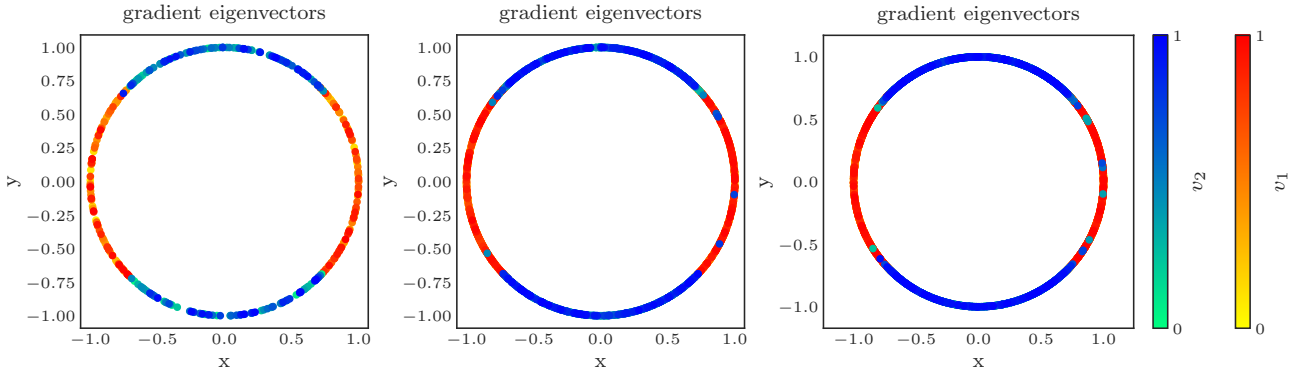
(a) 10 epochs (b) 50 epochs (c) 100 epochs

Figure 6. Distribution of eigenvectors of the gradient $\partial\mathcal{L}/\partial\mathbf{\Sigma}'_{2D}$ after 10, 50, and 100 epochs. Eigenvectors are color coded (green to blue and yellow to red) depending, whether there are the 1st or 2nd eigenvector and their epoch. While after 100 epochs most of the circle is covered, the eigenvectors aggregate at certain positions. Especially after 10 epochs, the eigenvectors are sparsely distributed. This shows a limited range of directions for the gradient.



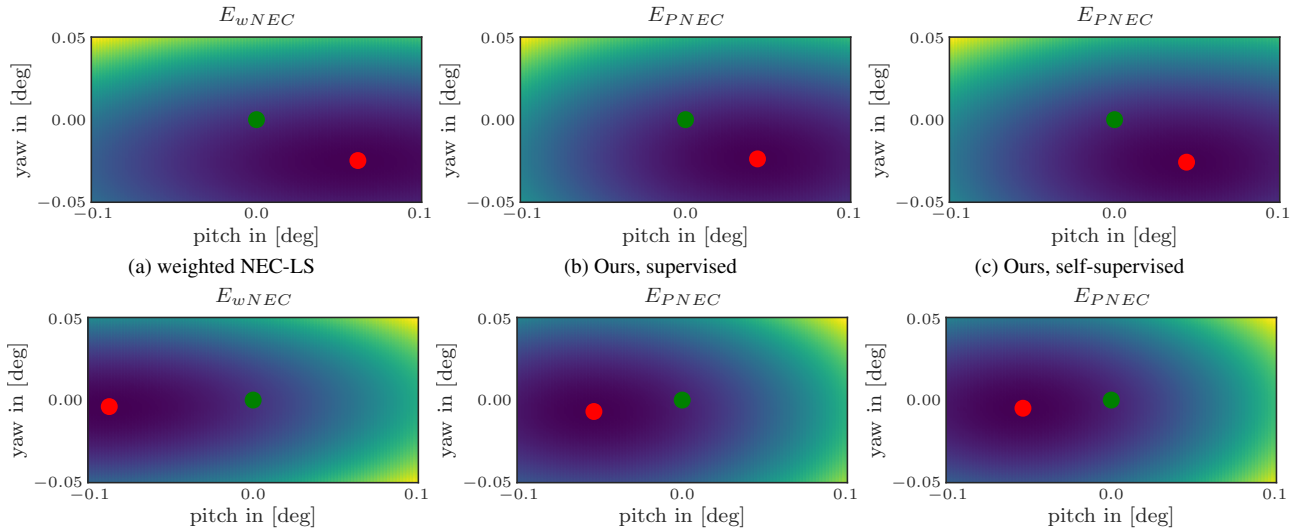
(a) 10 epochs (b) 50 epochs (c) 100 epochs

Figure 7. Histogram of eigenvector angles for the gradient $\partial\mathcal{L}/\partial\mathbf{\Sigma}'_{2D}$ after 10, 50, and 100 epochs. While it shows 4 distinct peaks, event after only 10 epochs many points lie in between. The direction of the gradient is not limited, allowing for a better fit to the ground truth noise distribution.



(a) 10 epochs (b) 50 epochs (c) 100 epochs

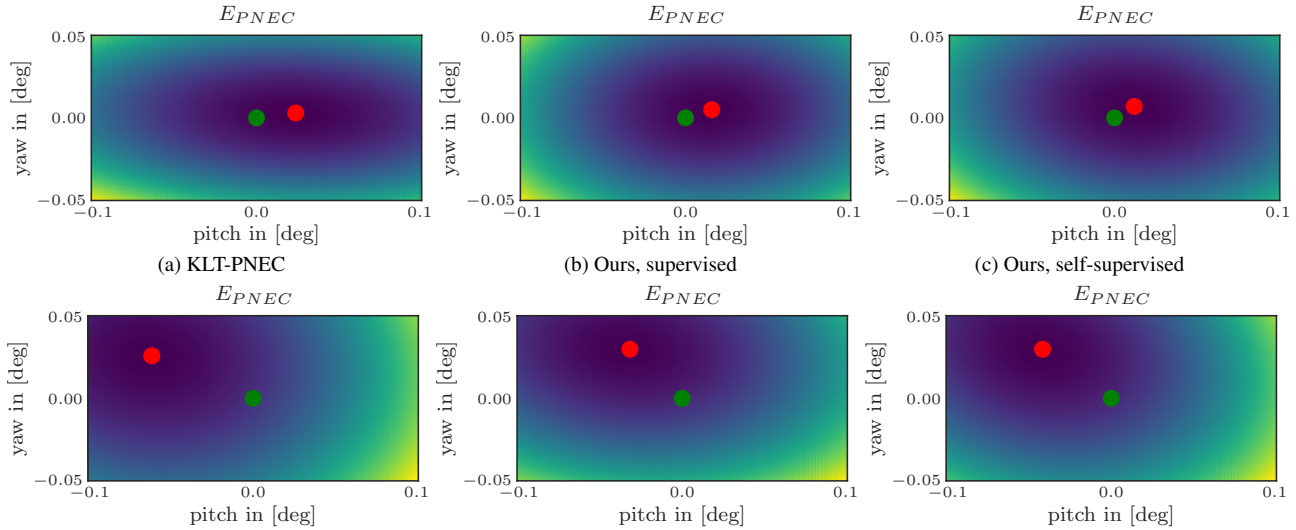
Figure 8. Distribution of eigenvectors of the gradient $\partial\mathcal{L}/\partial\mathbf{\Sigma}'_{2D}$ after 10, 50, and 100 epochs. Eigenvectors are color coded (green to blue and yellow to red) depending, whether there are the 1st or 2nd eigenvector and their epoch. Even after 10 epochs, the eigenvectors are evenly distributed. This show, that the gradient has no limit for its direction, allowing for a better fit to the noise distribution even in the image plane.



(a) weighted NEC-LS (b) Ours, supervised (c) Ours, self-supervised

(d) weighted NEC-LS (e) Ours, supervised (f) Ours, self-supervised

Figure 9. Energy functions evaluated for rotations around the ground truth pose (green). Minimum of the cost function is marked in red. The energy function is evaluated for SuperPoint keypoint for two pose estimation problems on the KITTI dataset, filtered with RANSAC at the ground truth pose. We compare the weighted NEC-LS energy function to the PNEC energy function with our supervised and self-supervised covariances. While the overall shape of the energy function stays the same, our learned covariances move the minimum closer to the ground truth.



(a) KLT-PNEC (b) Ours, supervised (c) Ours, self-supervised

(d) KLT-PNEC (e) Ours, supervised (f) Ours, self-supervised

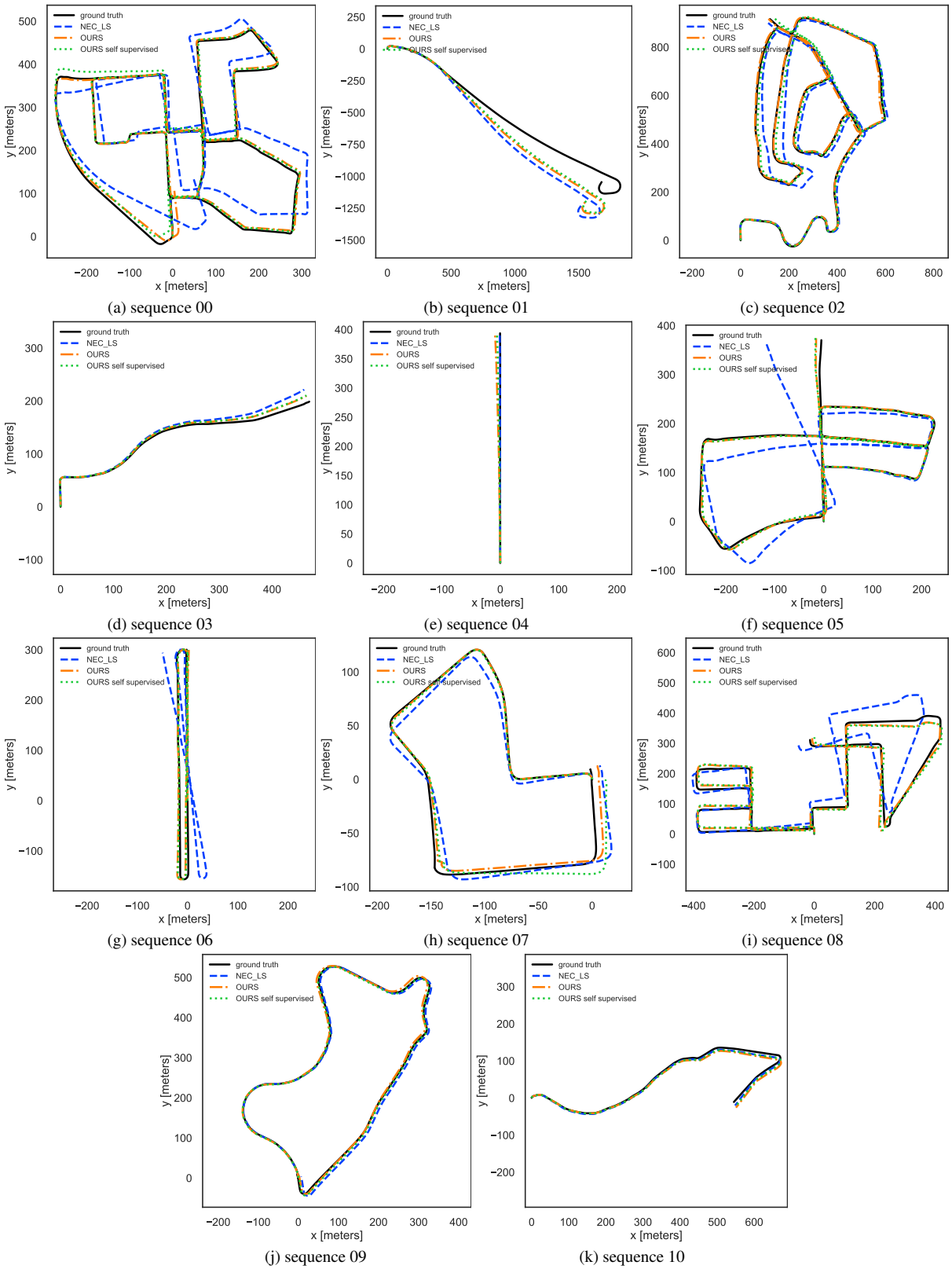
Figure 10. Energy functions evaluated for rotations around the ground truth pose (green). Minimum of the cost function is marked in red. The energy function is evaluated for KLT-tracks for two pose estimation problems on the KITTI dataset, filtered with RANSAC at the ground truth pose. We compare the PNEC energy function using the KLT-covariances with our supervised and self-supervised covariances. While the overall shape of the energy function stays the same, our learned covariances move the minimum closer to the ground truth.

	8PT [5]			NEC [4]			NEC-LS			OURS SUPERVISED			OURS SELF-SUPERVISED			REPROJECTION		
Seq.	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t
00	0.185	7.203	2.61	0.153	5.505	9.32	0.121	**2.403**	1.42	0.115	2.994	1.31	**0.113**	3.110	**1.30**	0.117	3.080	1.29
01	**0.253**	7.162	2.89	0.659	28.523	5.24	0.270	8.991	**2.20**	0.294	6.433	2.23	0.349	**6.042**	2.27	0.363	7.712	2.20
02	0.159	7.451	1.85	0.115	6.891	7.69	0.079	3.751	1.06	0.078	3.411	0.99	**0.075**	**3.342**	**0.99**	0.083	4.410	0.99
03	0.131	4.822	2.47	0.089	1.889	7.45	0.051	1.493	1.17	0.058	0.602	1.01	**0.049**	**0.444**	**1.00**	0.047	0.608	1.00
04	0.126	1.899	1.08	0.037	0.846	6.42	0.037	0.816	0.50	**0.030**	**0.387**	0.44	0.030	0.428	**0.43**	0.028	0.549	0.33
05	0.148	5.563	3.35	0.155	10.630	9.75	0.089	6.352	2.40	**0.046**	1.285	**2.23**	0.046	**1.235**	2.23	0.056	1.644	2.17
06	0.142	3.376	1.55	0.066	1.984	7.30	0.044	**1.325**	0.63	0.032	1.576	**0.50**	**0.032**	1.569	0.50	0.031	1.467	0.45
07	0.170	5.347	6.41	0.258	12.558	12.51	0.120	5.371	5.58	**0.094**	2.731	**4.97**	0.098	**2.500**	5.15	0.073	2.132	4.18
08	0.144	8.508	3.49	0.088	3.902	8.91	0.053	2.908	2.49	0.048	2.373	**2.36**	**0.047**	**1.706**	2.36	0.047	2.454	2.31
09	0.151	4.546	1.71	0.054	2.027	6.76	0.052	2.307	0.74	0.043	1.244	**0.64**	**0.042**	**1.141**	0.64	0.044	1.385	0.64
10	0.148	6.540	2.88	0.119	8.302	8.53	0.066	4.576	1.78	0.058	3.789	**1.58**	**0.056**	**3.623**	1.60	0.057	2.615	1.37
train	0.168	6.407	2.69	0.173	8.301	8.59	0.103	3.955	1.73	**0.094**	2.782	**1.60**	0.096	**2.737**	1.61	0.100	3.193	1.52
test	0.146	7.246	2.97	0.085	4.237	8.34	0.055	3.060	1.96	0.048	2.359	**1.82**	**0.048**	**1.910**	1.83	0.048	2.234	1.76

Table 3. Quantitative comparison on the KITTI [2] dataset with Kanade-Lucas-Tomasi (KLT) tracks [9]. We replace the Nistér-5pt [7] with the 8pt [5] algorithm to show more results. We also show, an approximation of the true error distance using reprojected points (this is excluded from being **bold** or underlined). While the reprojection approximation achieves the best results on almost all sequences, our methods are often not far behind. This emphasises, that our method is able to effectively learn covariances.

	8PT [5]			NEC [4]			NEC-LS			OURS SUPERVISED			OURS SELF-SUPERVISED			REPROJECTION		
Seq.	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t
00	0.216	11.650	3.40	0.132	12.483	3.20	0.116	8.728	**1.35**	0.114	**2.277**	1.38	**0.114**	2.522	1.38	0.113	2.363	1.28
01	0.246	8.080	3.83	0.539	22.857	1.55	0.082	6.378	1.00	0.060	5.811	0.99	**0.057**	**5.770**	**0.94**	0.054	5.997	0.81
02	0.188	12.003	2.06	0.093	7.594	1.76	0.069	4.050	1.01	0.066	**2.224**	**0.99**	**0.066**	2.237	1.00	0.065	2.679	0.95
03	0.167	8.308	3.42	0.090	3.863	3.31	**0.055**	3.754	1.12	0.059	2.239	1.13	0.057	**2.051**	**1.12**	0.054	2.394	1.07
04	0.160	2.682	1.45	0.040	0.486	0.81	0.041	**0.434**	0.49	0.038	1.041	**0.46**	**0.037**	0.808	0.46	0.027	0.526	0.30
05	0.198	9.236	4.56	0.119	11.779	3.65	0.062	12.437	2.50	0.055	**1.931**	**2.37**	**0.055**	1.949	2.40	0.053	2.123	2.02
06	0.193	5.244	2.89	0.059	6.901	1.43	0.050	6.634	0.76	0.042	**1.178**	0.70	**0.041**	1.242	**0.70**	0.035	0.964	0.58
07	0.231	7.086	8.86	0.185	4.402	8.67	0.112	**2.341**	6.69	**0.103**	2.772	**6.54**	0.109	3.715	6.63	0.120	3.434	4.82
08	0.183	10.423	4.21	0.081	8.284	3.66	0.056	7.004	2.50	**0.050**	**4.067**	2.46	0.050	4.118	**2.46**	0.048	3.623	2.30
09	0.185	5.485	2.29	0.053	1.646	1.43	0.052	1.553	0.71	0.049	1.317	0.71	**0.049**	**1.278**	**0.70**	0.048	1.160	0.69
10	0.198	8.960	4.09	0.167	9.264	4.43	0.064	4.787	1.79	**0.063**	**3.513**	**1.64**	0.065	3.821	1.65	0.060	2.404	1.21
train	0.203	10.051	3.54	0.141	10.127	2.97	0.082	6.910	1.72	0.077	**2.378**	**1.69**	**0.077**	2.505	1.69	0.076	2.606	1.44
test	0.186	9.023	3.74	0.089	6.917	3.28	0.056	5.353	1.96	**0.052**	**3.333**	1.91	0.053	3.408	**1.91**	0.050	2.839	1.73

Table 4. Full results on the KITTI [2] dataset with SuperPoint [1] keypoints. We replace the Nistér-5pt [7] with the 8pt [5] algorithm to show more results. We also show, an approximation of the true error distance using reprojected points (this is excluded from being **bold** or underlined). While the reprojection approximation achieves the best results on almost all sequences, our methods are often not far behind. This emphasises, that our method is able to effectively learn covariances.



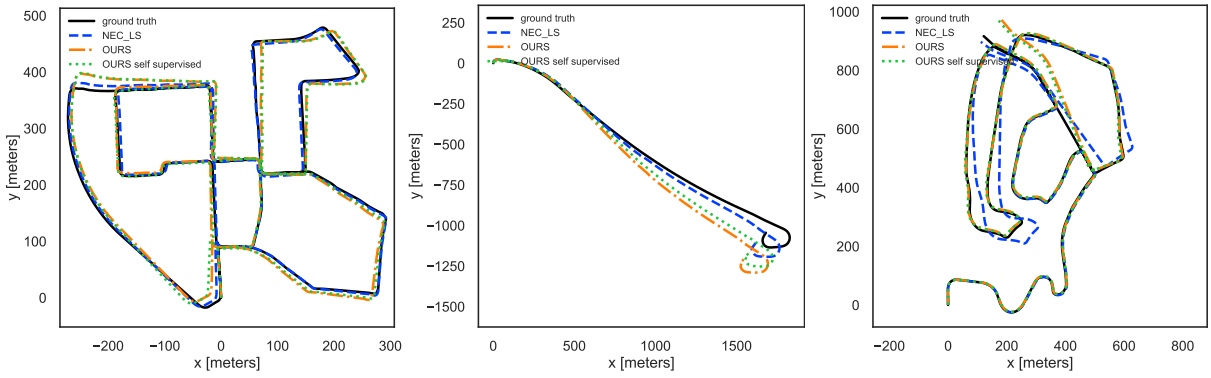
(a) sequence 00 (b) sequence 01 (c) sequence 02

(d) sequence 03 (e) sequence 04 (f) sequence 05

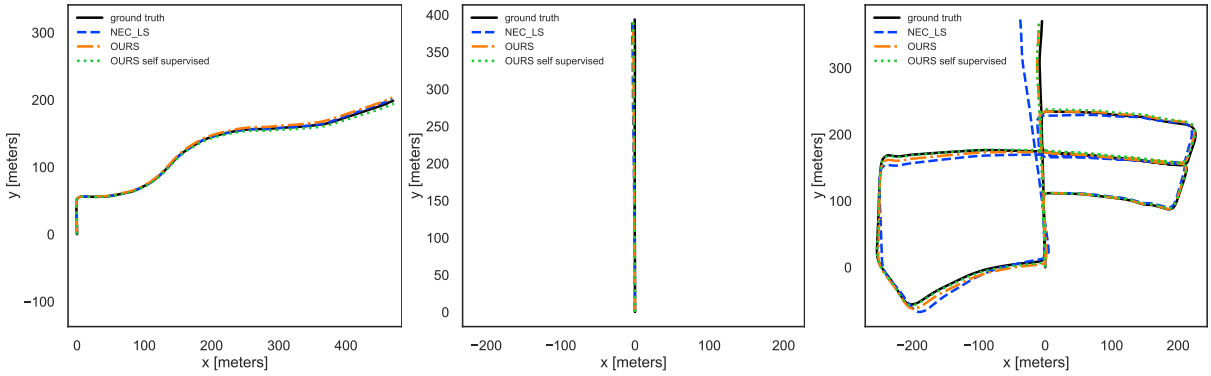
(g) sequence 06 (h) sequence 07 (i) sequence 08

(j) sequence 09 (k) sequence 10

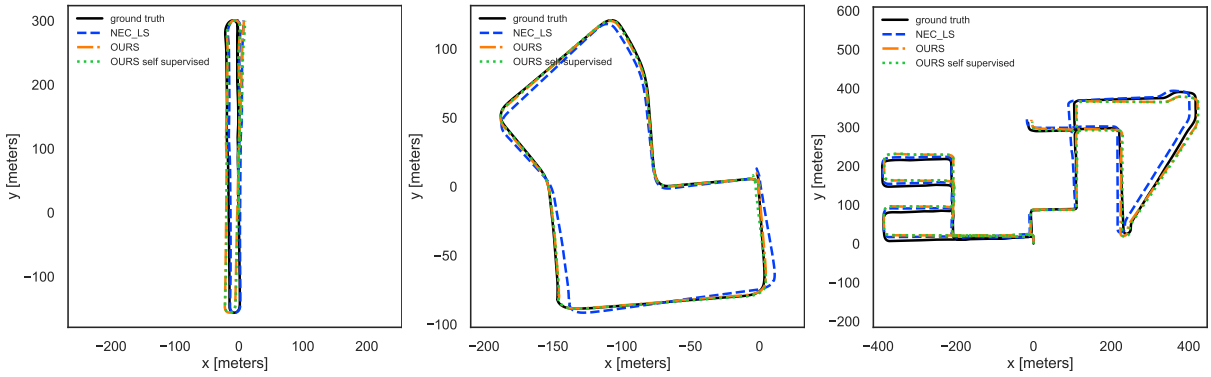
Figure 11. Trajectory comparison for the KITTI visual odometry sequences for SuperPoint keypoints. Since we compare monocular methods, that cannot estimate the correct scale from a pair of images, we use the scale of the ground truth translations for visualization purposes.



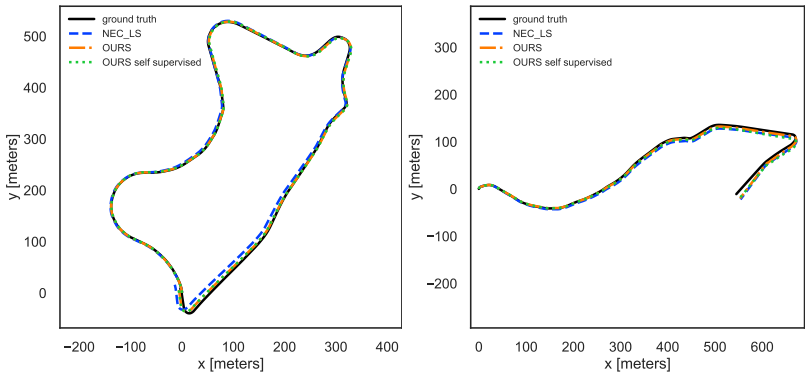
(a) sequence 00 (b) sequence 01 (c) sequence 02



(d) sequence 03 (e) sequence 04 (f) sequence 05



(g) sequence 06 (h) sequence 07 (i) sequence 08



(j) sequence 09 (k) sequence 10

Figure 12. Trajectory comparison for the KITTI visual odometry sequences for KLT-tracks. Since we compare monocular methods, that cannot estimate the correct scale from a pair of images, we use the scale of the ground truth translations for visualization purposes.

References

[1] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018. 7

[2] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012. 1, 7

[3] Laurent Kneip and Paul Furgale. Opengv: A unified and generalized approach to real-time calibrated geometric vision. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. 4

[4] Laurent Kneip and Simon Lynen. Direct optimization of frame-to-frame rotation. In *ICCV*, 2013. 7

[5] HC Longuet-Higgins. Readings in computer vision: issues, problems, principles, and paradigms. *A computer algorithm for reconstructing a scene from two projections*, 1987. 4, 7

[6] D Muhle, L Koestler, N Demmel, F Bernard, and D Cremers. The probabilistic normal epipolar constraint for frame-to-frame rotation optimization under uncertain feature positions. 2022. 4

[7] D. Nister. An efficient solution to the five-point relative pose problem. In *CVPR*, 2003. 7

[8] D. Nistr, O. Naroditsky, and J. Bergen. Visual odometry. *CVPR*, 2004. 4

[9] Vladyslav Usenko, Nikolaus Demmel, David Schubert, Jörg Stückler, and Daniel Cremers. Visual-inertial mapping with non-linear factor recovery. *IEEE Robotics and Automation Letters (RAL)*, 5, 2020. 7