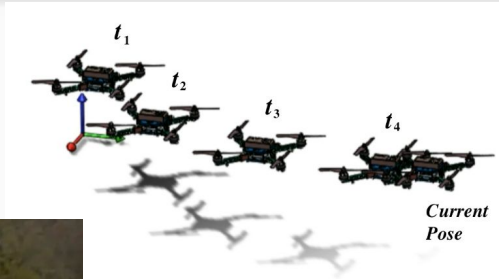
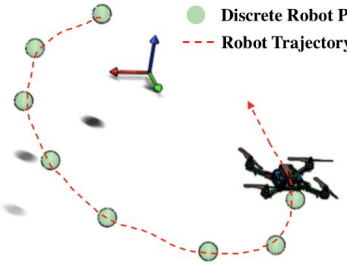
Deep Learning for Visual Odometry

Vladislav Trifonov
Hekmat Taherinejad
Prateek Rajput
Vladimir Chernyy
Anna Akhmatova
Timur Bayburin

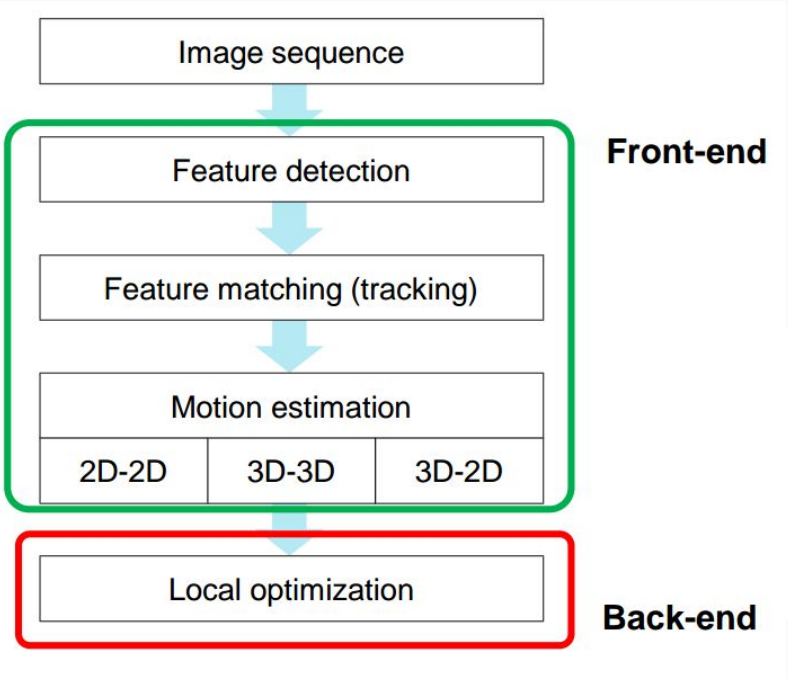
Deep Learning 2022 course

Introduction

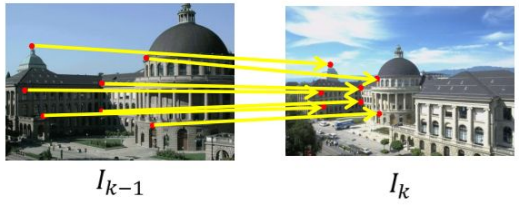


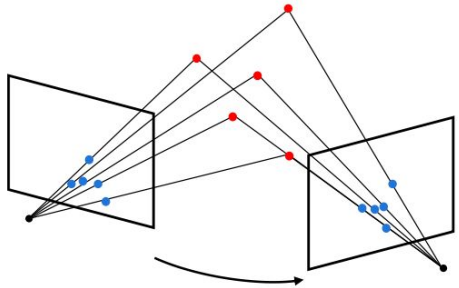


Literature review



- **Traditional Visual Odometry (ORB SLAM)**
- **Depth-Pose Learning without PoseNet**
- **Transformers with attention blocks**





Aim of the project

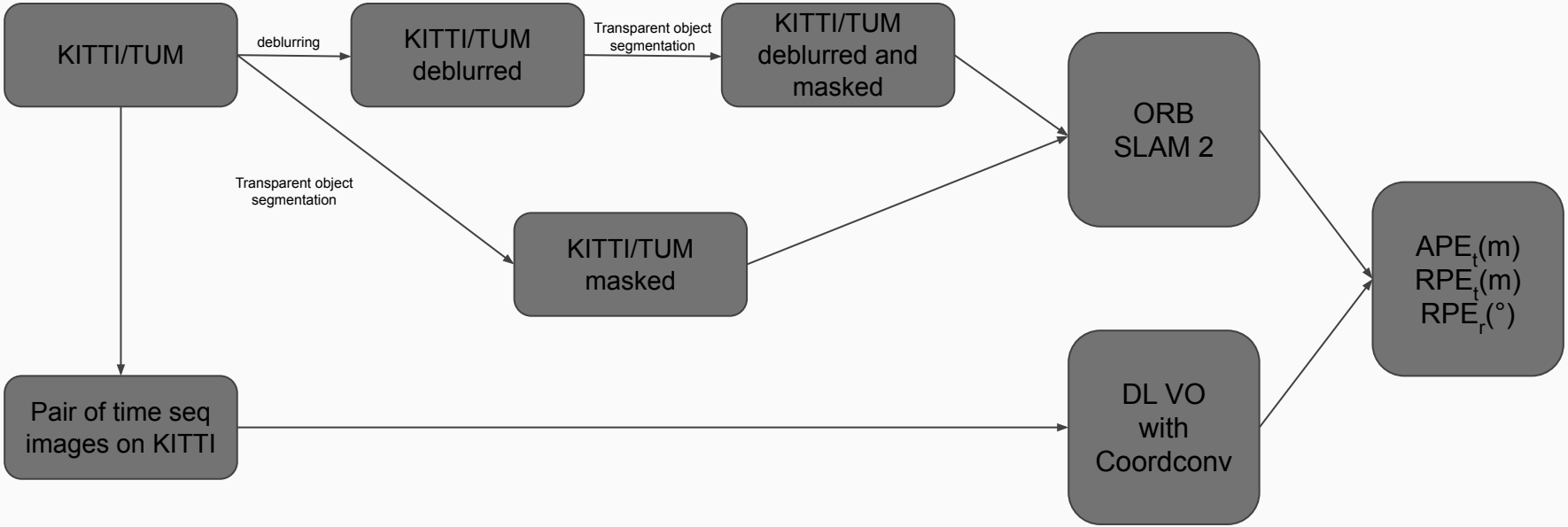
The are several **concerns** regarding current state-of-the-art:

- Few attention is given to **deep learning methods** in Frontends steps for obtaining better accuracy and faster results.
- The idea of implementing the whole pipeline of Visual Odometry based on learning methods is still under research and exploration.
- Few studies has been made towards **preprocessing visual inputs by CNN** for sharper better quality images.

Objectives

- To implement a **fully connected** Visual Odometry just by using deep learning methods and train a network for this purpose.
- To implement effective deep learning methods for **enhancing ORB SLAM 2** Visual Odometry pipeline.
- To **compare** fully deep learning Visual Odometry method with ORB SLAM 2.

Methodology



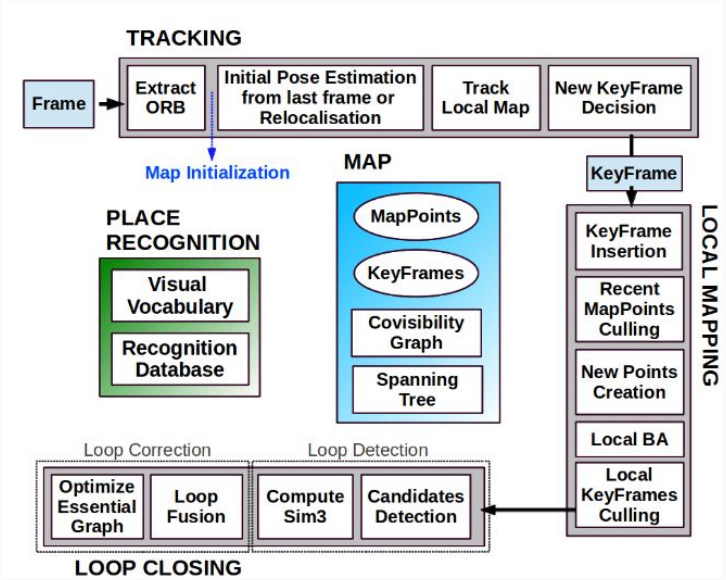
ORB SLAM2

ORB-SLAM2 - real-time SLAM library

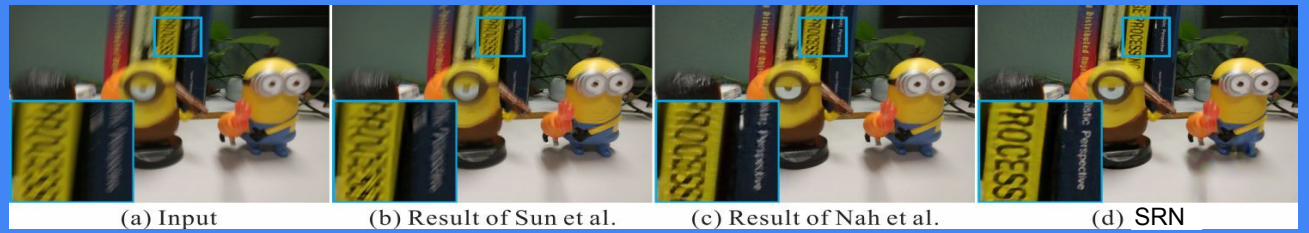
Proceeds: **Monocular**, **Stereo**, **RGB-D**

Computes camera **trajectory** and sparse 3D reconstruction (in the stereo and RGB-D case with true scale)

It is able to detect **loops** and relocalize the camera in real time

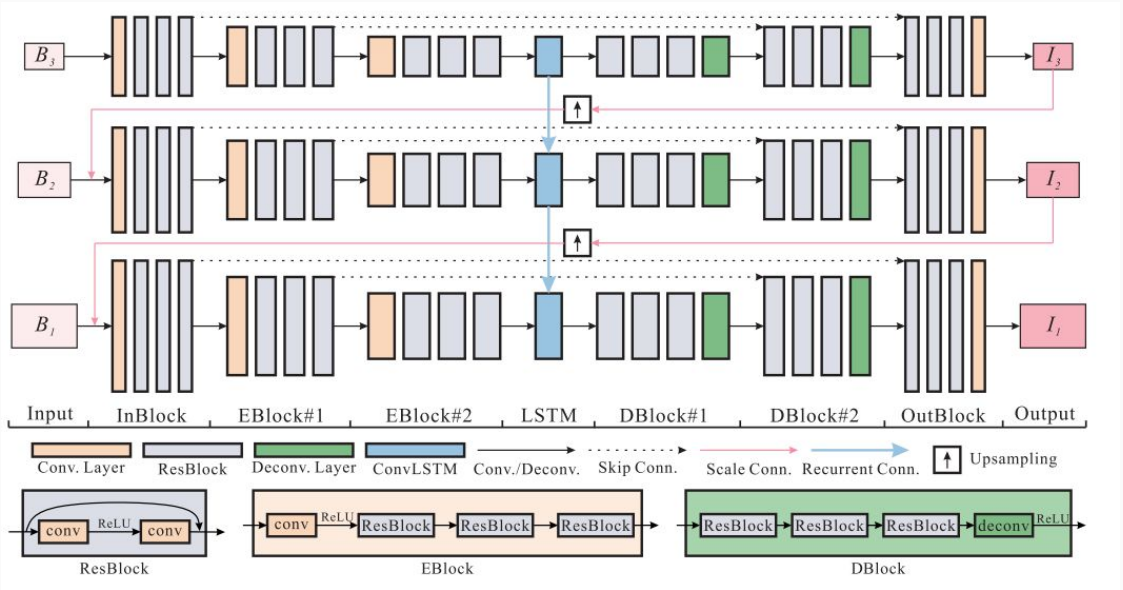


Deblurring

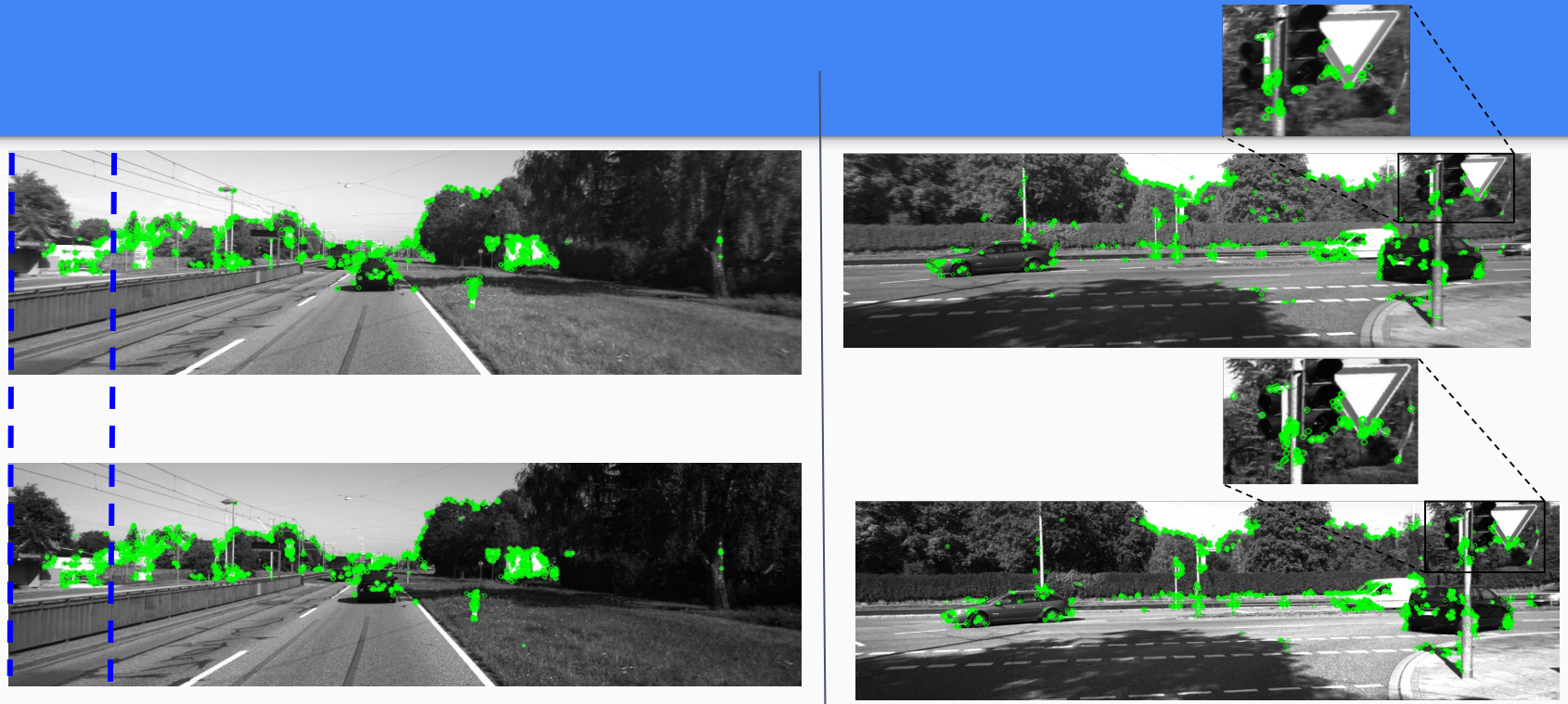


- Train the network using the GOPRO dataset.

Method	PSNR	SSIM
Kim et al.	23.71	0.815
Sun et al.	24.75	0.832
Nah et al.	28.91	0.921
SRN-DeblurNet	**30.15**	**0.942**



Deblurring

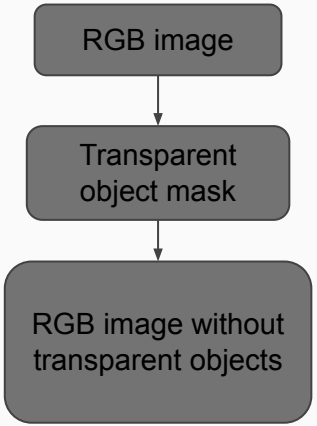


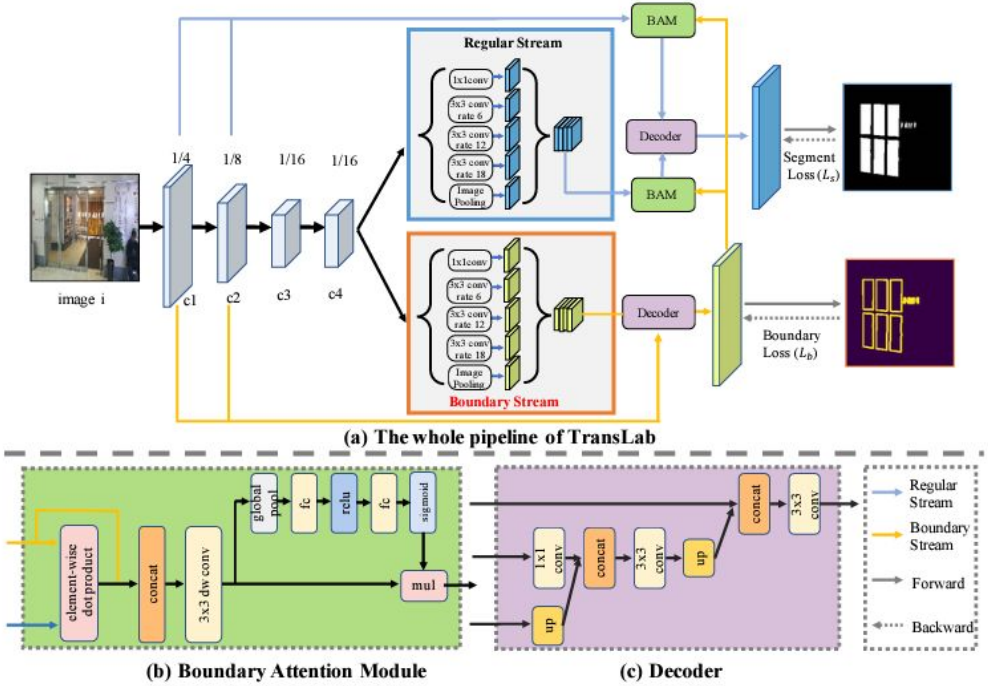
Kitti dataset

Transparent object masking

CNN network plus Translab architecture combination for boundary aware object segmentation

The network is trained on Trans 10k dataset by dividing the images into easy hard training datasets



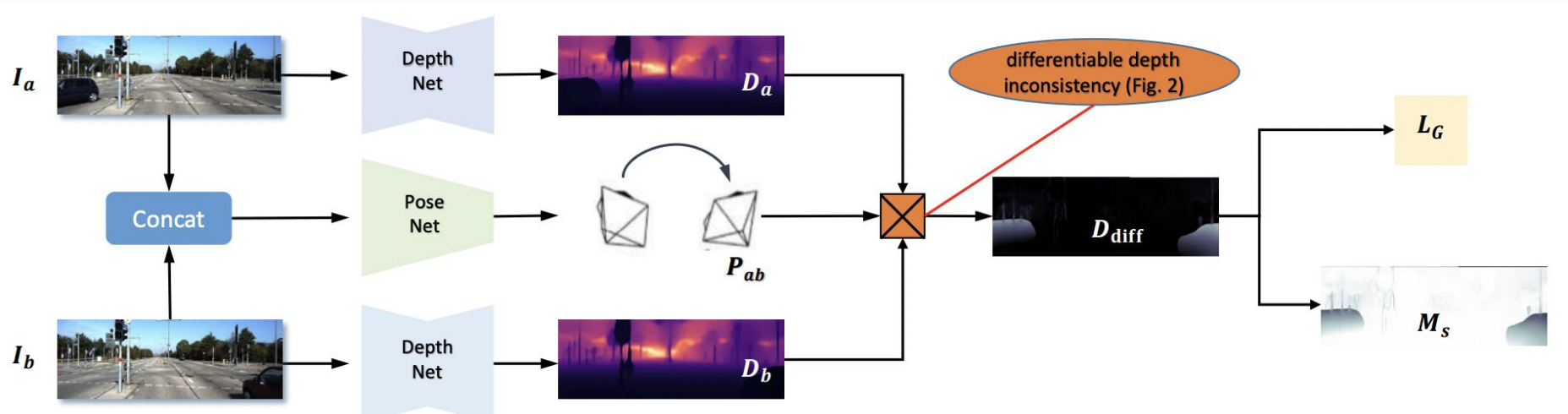


Examples of masked out images

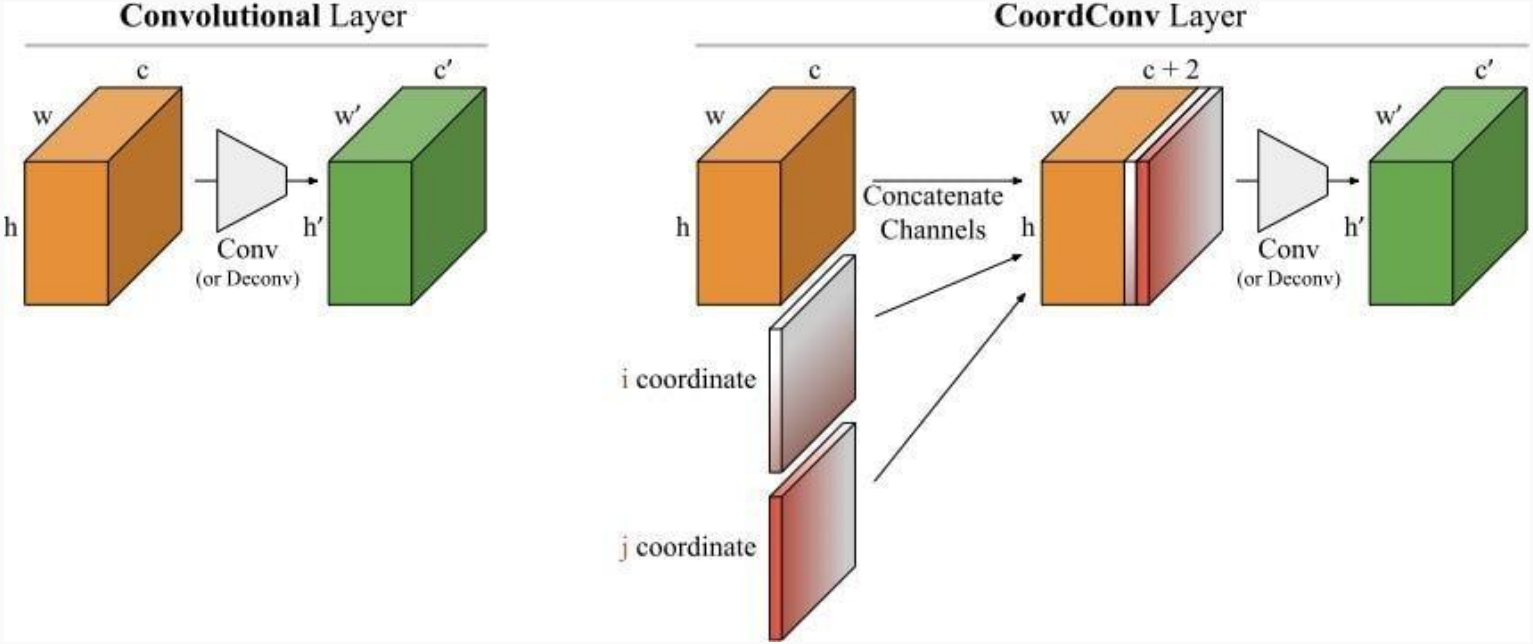
Deblurred and transparent object segmented image from KITTI seq 8

Deblurred and transparent object segmented image from TUM r3/long office household

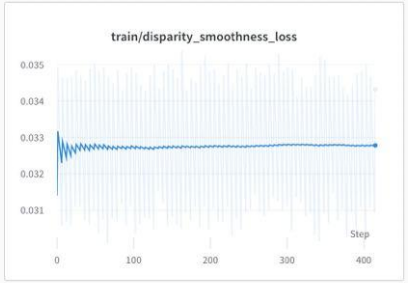
PoseNet with CoordConv

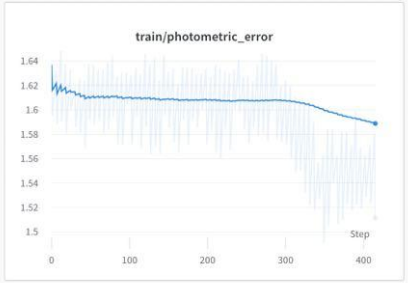


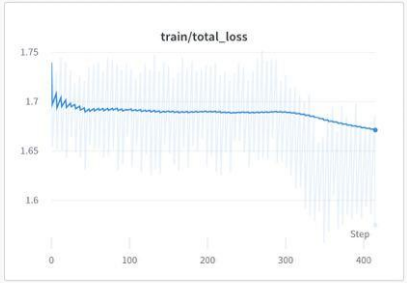
PoseNet with CoordConv

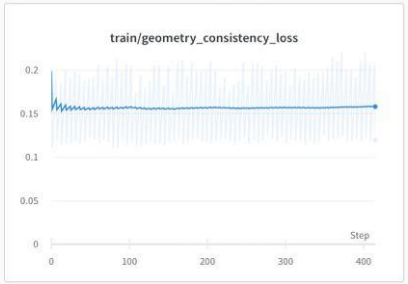


Network overfitting

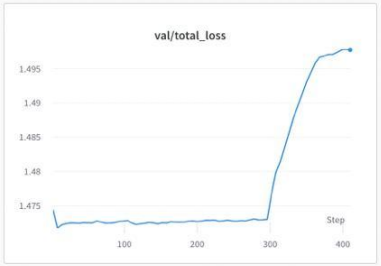


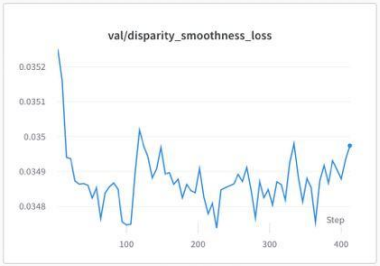


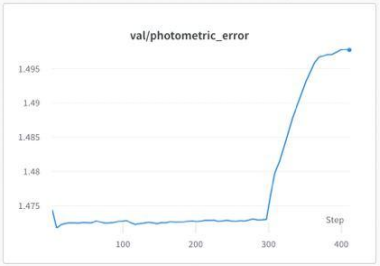


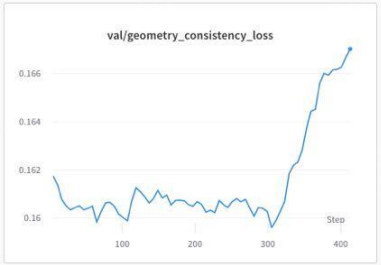


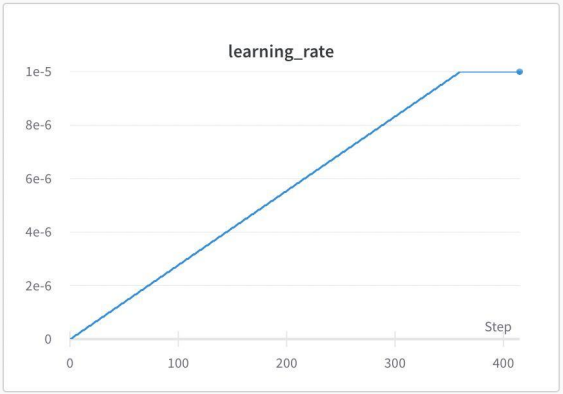
Network overfitting











Implementation

Datasets:

1. **KITTI** odometry sequences: 08, 09, 10
2. **TUM** sequences:

   freiburg1_rpy, freiburg1_xyz, freiburg3_long_office_household (includes transparents)

Evaluation Metrics:

APE - Absolute Pose Error is determined by comparing the predicted and GT trajectories absolute distances. $E_i = P_{ref,i}^{-1} P_{est,i} \in \mathrm{SE}(3)$

RPE - Relative Pose Error is a measurement of the trajectories local accuracy over a fixed time interval Delta. $E_{i,j} = (P_{ref,i}^{-1} P_{ref,j})^{-1} (P_{est,i}^{-1} P_{est,j}) \in \mathrm{SE}(3)$

Results

Metrics:
t - translation part
r - rotation part

Sequences:
o - original
s - segmented
d - deblurred

TUM

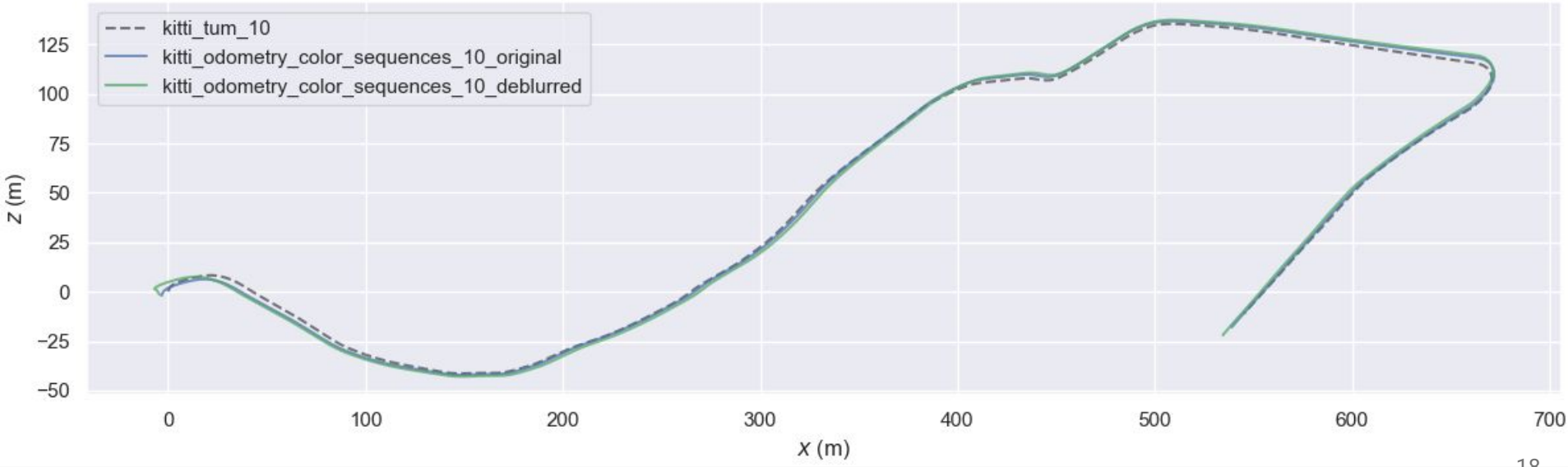
	fr1/rpy			fr1/xyz			fr3/long		
	$APE_t(m)$	$RPE_t(m)$	$RPE_r(°)$	$APE_t(m)$	$RPE_t(m)$	$RPE_r(°)$	$APE_t(m)$	$RPE_t(m)$	$RPE_r(°)$
o	0.024508	0.018725	1.444773	**0.005638**	**0.008801**	**0.544655**	0.031001	0.007968	0.366255
s	0.022390	**0.013729**	1.280872	0.006539	0.010742	0.647860	**0.025041**	0.007128	**0.360615**
d	**0.018767**	0.017813	**1.127863**	0.008820	0.013572	0.696928	0.034174	0.007442	0.336747
d+s	**0.018613**	0.065011	2.190500	0.007662	0.012044	0.710171	0.025793	**0.006862**	0.336524

KITTI

	seq08			seq09			seq10		
	$APE_t(m)$	$RPE_t(m)$	$RPE_r(°)$	$APE_t(m)$	$RPE_t(m)$	$RPE_r(°)$	$APE_t(m)$	$RPE_t(m)$	$RPE_r(°)$
o	**37.757439**	**0.467190**	**0.053345**	**31.126846**	0.635913	**0.053478**	**3.502739**	**0.060417**	**0.059422**
d	38.701056	0.487801	**0.053924**	42.403437	**0.334358**	**0.053853**	5.366431	0.095038	0.072026

Result trajectories

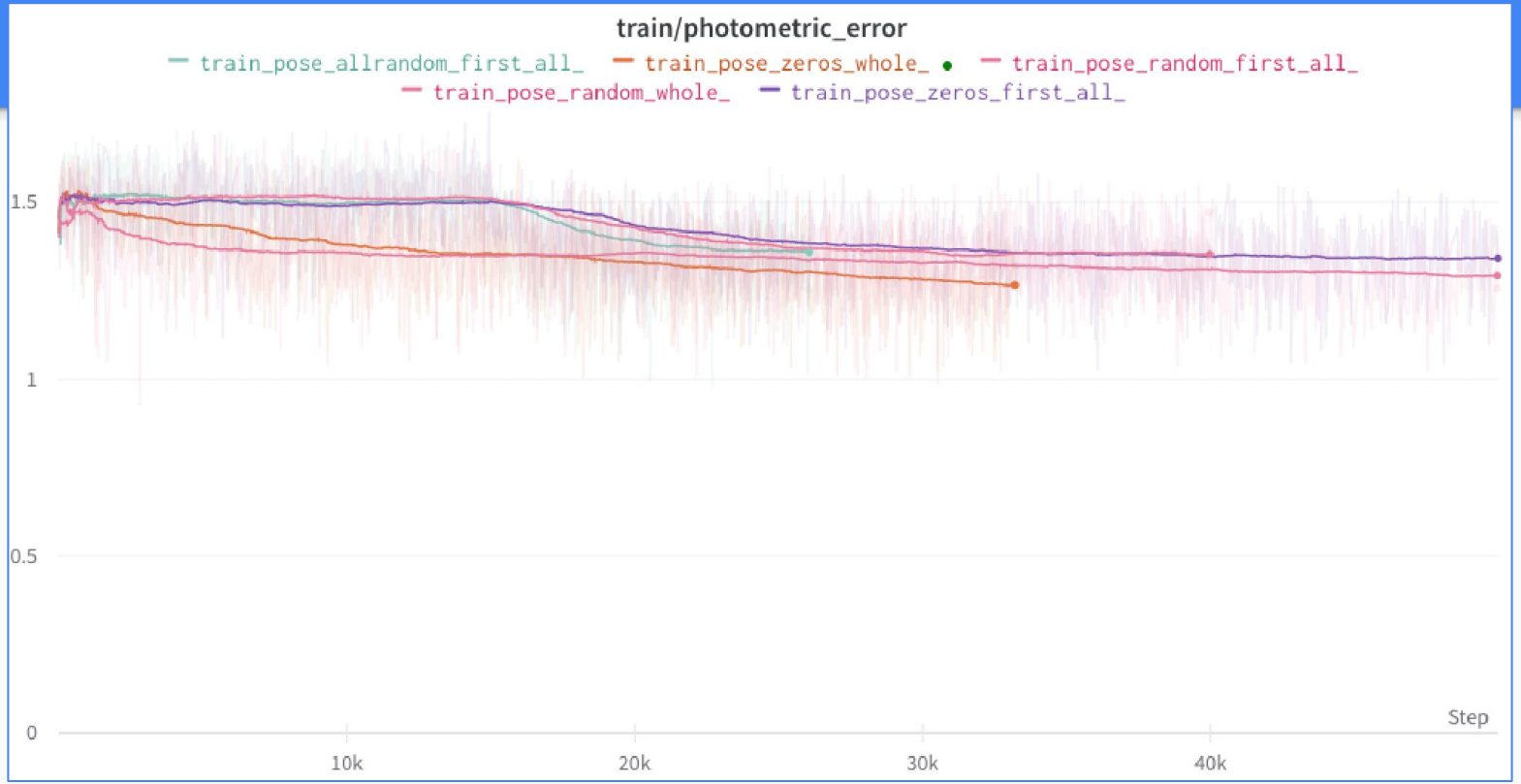
KITTI sequence 10



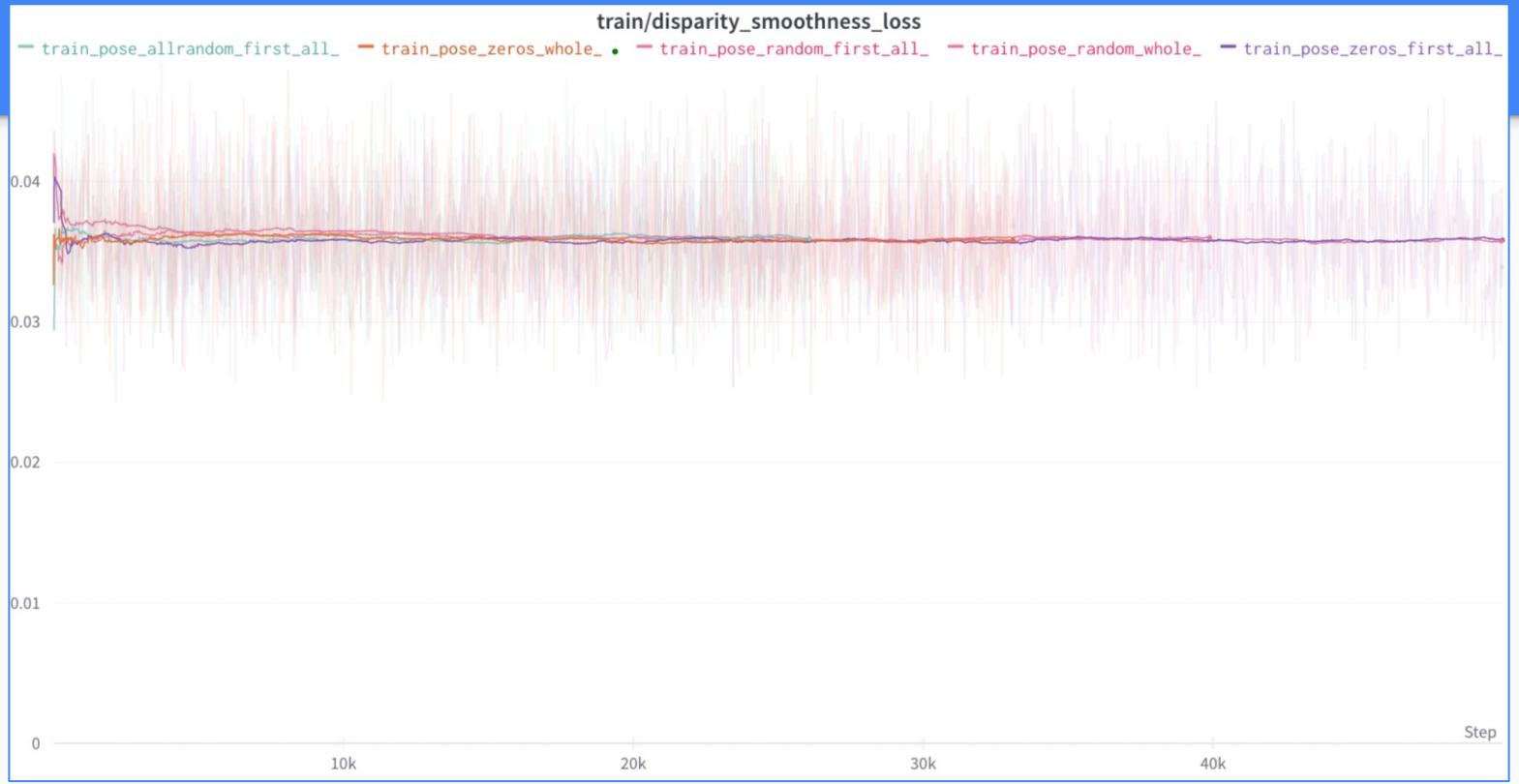
Translation and Rotation Errors

	seq09		seq10	
	t_err, %	r_err, (deg/100m)	t_err, %	r_err, (deg/100m)
source paper	**7.31**	**3.05**	**7.79**	**4.90**
random, first_all	86.94	24.92	147.39	22.77
zeros, whole	133.56	12.73	162.55	18.52
random, whole	92.36	25.25	147.66	22.45
allrandom, first_all	86.54	26.15	147.28	23.90
zeros, first_all	87.08	25.09	147.54	22.70

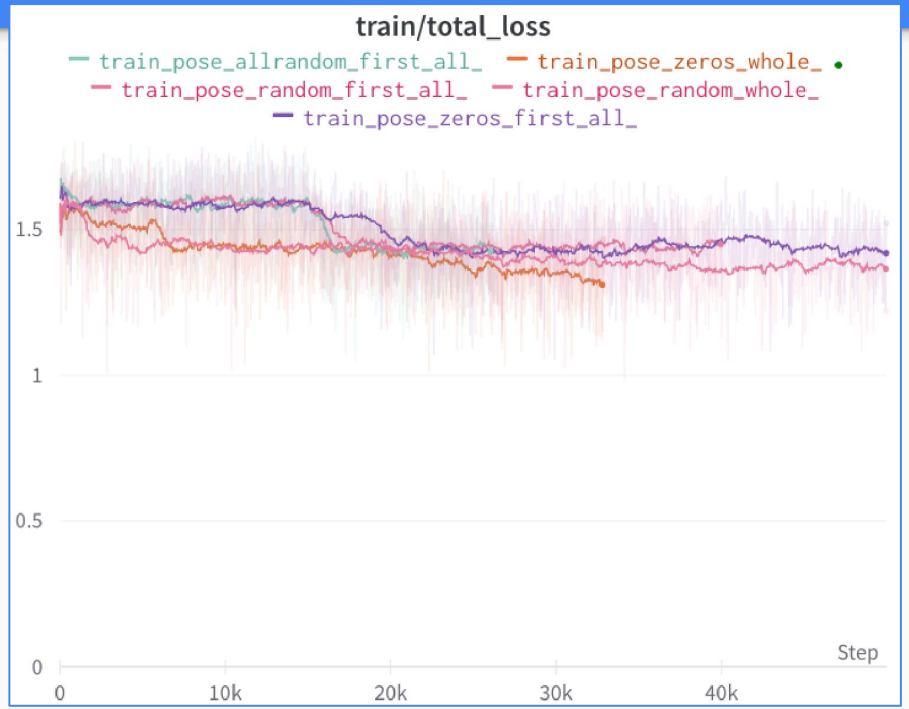
Training losses

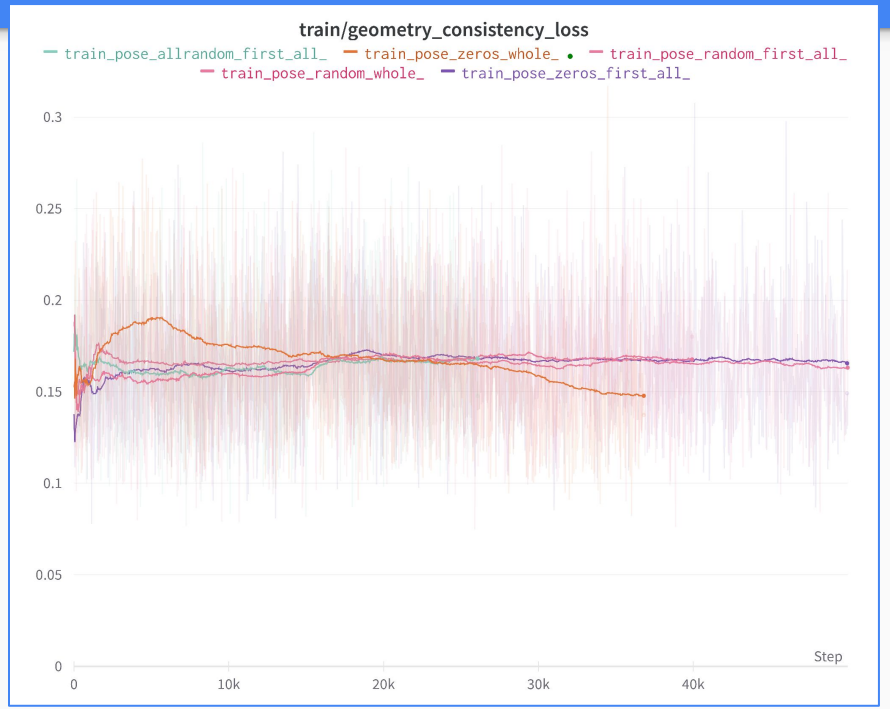


Training losses

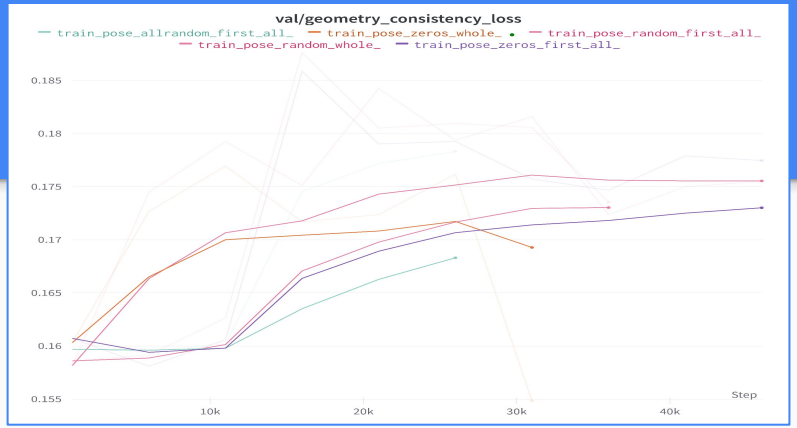


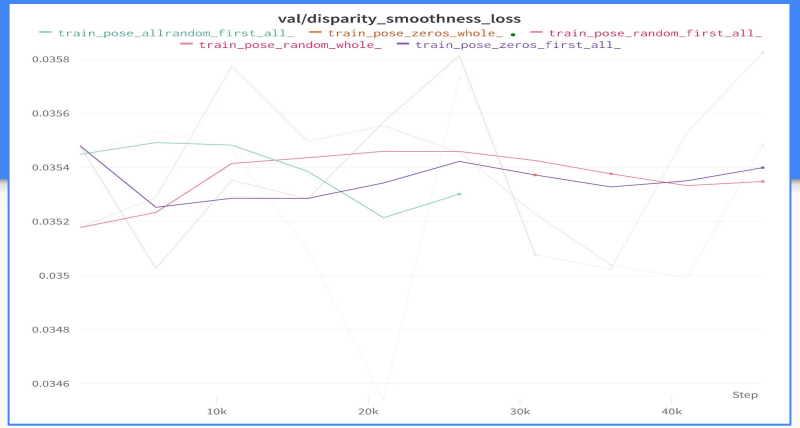
Training losses

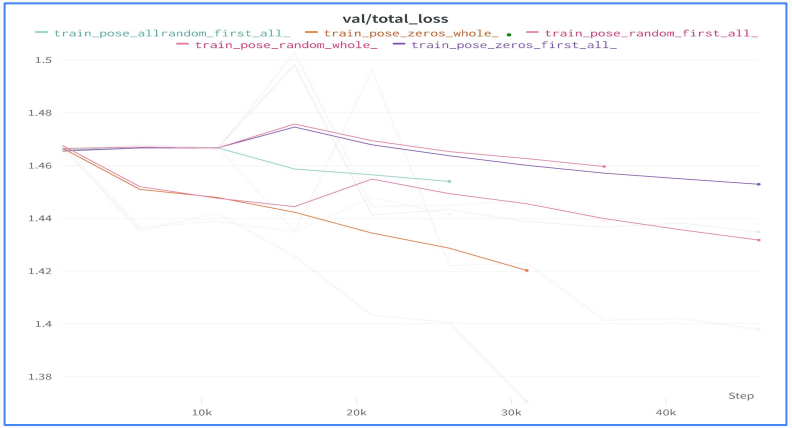


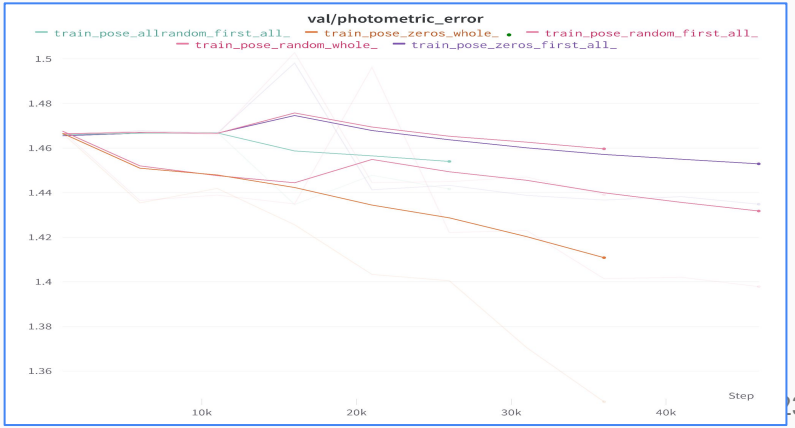


Validation losses









Conclusion

- In contrast to our expectations the deblurring preprocess didn’t enhance the performance of Visual Odometry always, and it added a remarkable amount of extra running time to the whole pipeline.
- For transparent object masking step, we realized that the network that we used didn’t work on outdoor dataset (Kitti) and on the indoor dataset (TUM) the metrics didn’t suggest a reliable improvement.
- The Deep Learning for Visual Odometry with CoordConv need more further investigation.