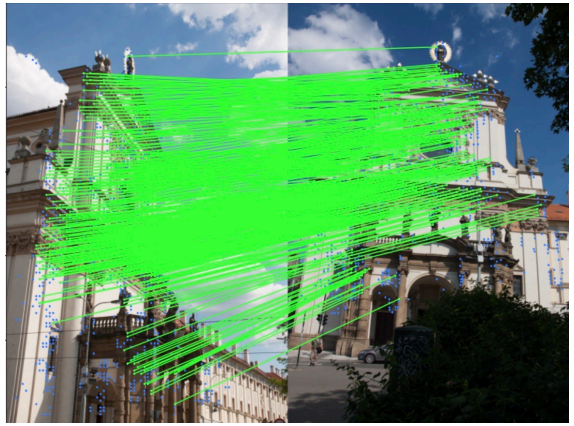
Learning Transformation-Predictive Representations for Detection and Description of Local Features

Zihao Wang, Chunxu Wu, Yifei Yang, Zhen Li

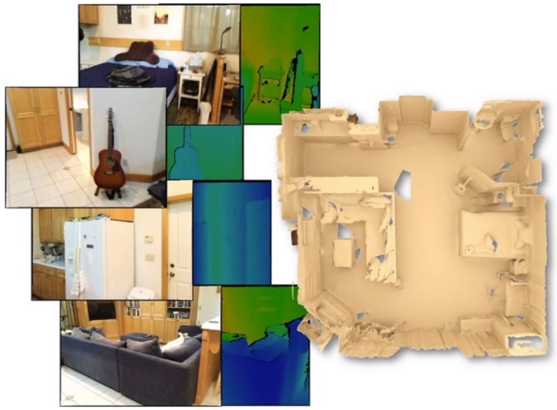
CVPR 2023

Background

Local visual descriptors are fundamental to various computer vision applications such as **camera calibration**, **3D reconstruction**, **vSLAM**, and **image retrieval**.

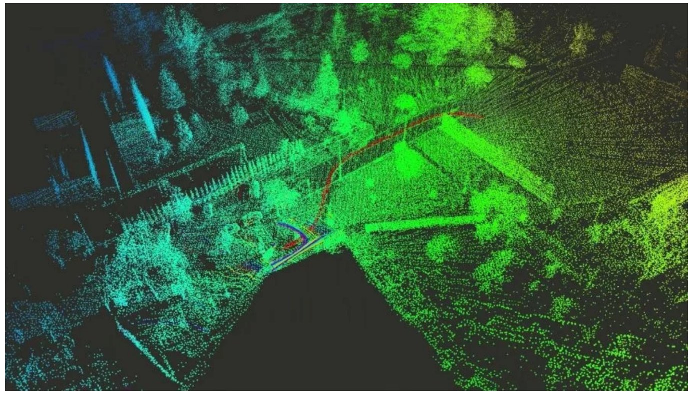


[Image Matching Challenge]



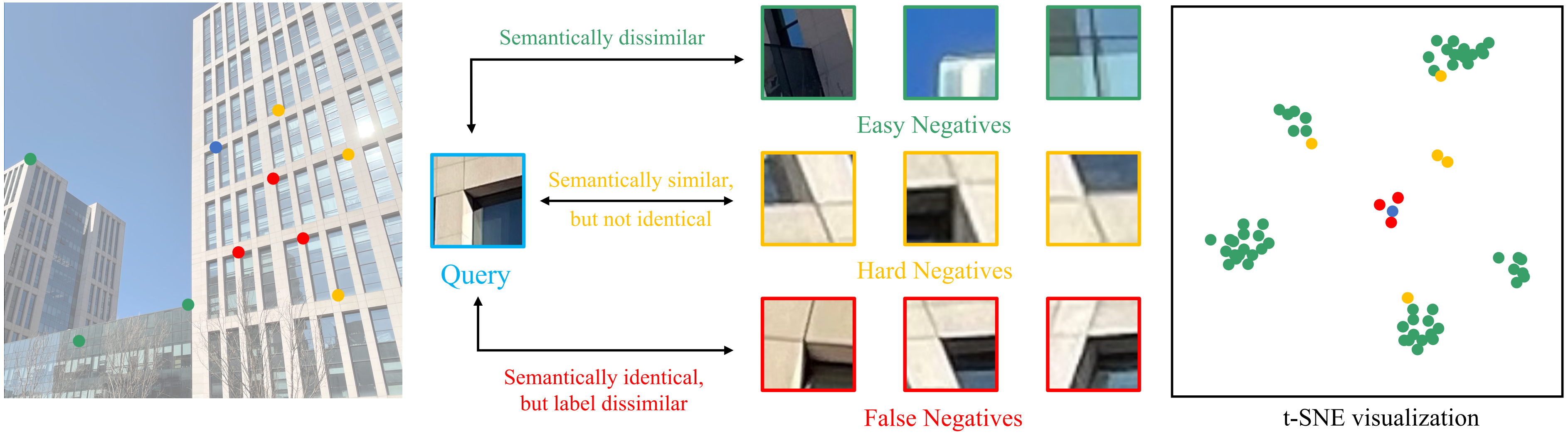
[ScanNet]

[Long-Term Visual Localization]



[SLAM]

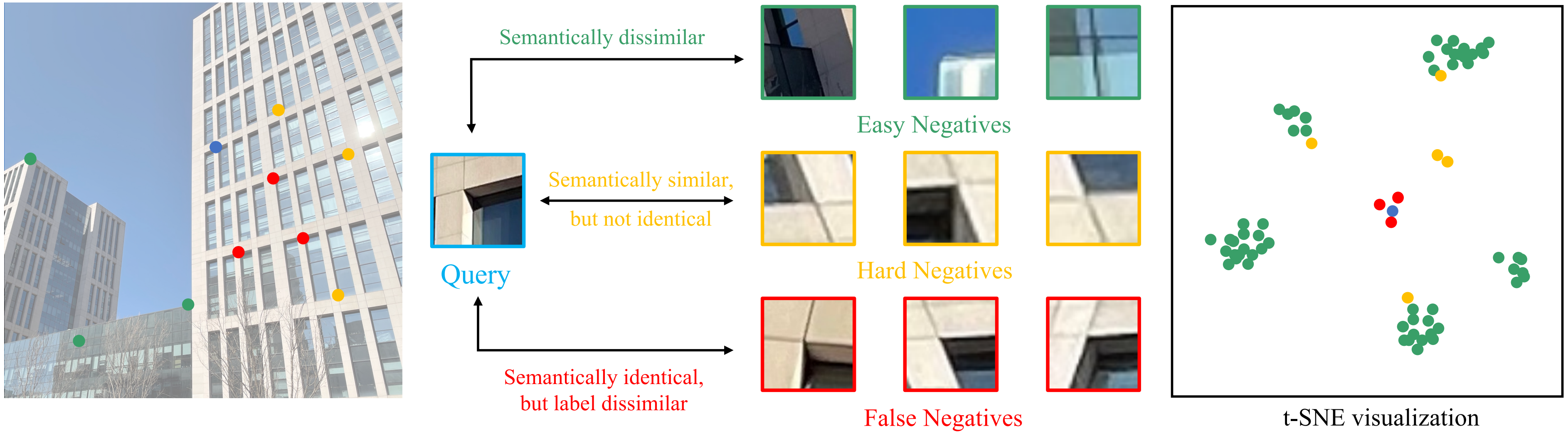
Motivation and Contribution



- **Motivation I**:
  - Negative samples are introduced to contrastive learning to keep the uniformity and avoid model collapse, while raise the computational load and memory usage heavily.
  - Some false negatives are labeled with hard negatives, leading to inconsistent supervision.
- **Contribution I**:

  we propose to learn transformation-predictive representations for joint local feature learning, using none of the negative sample pairs and avoiding collapsing solutions.

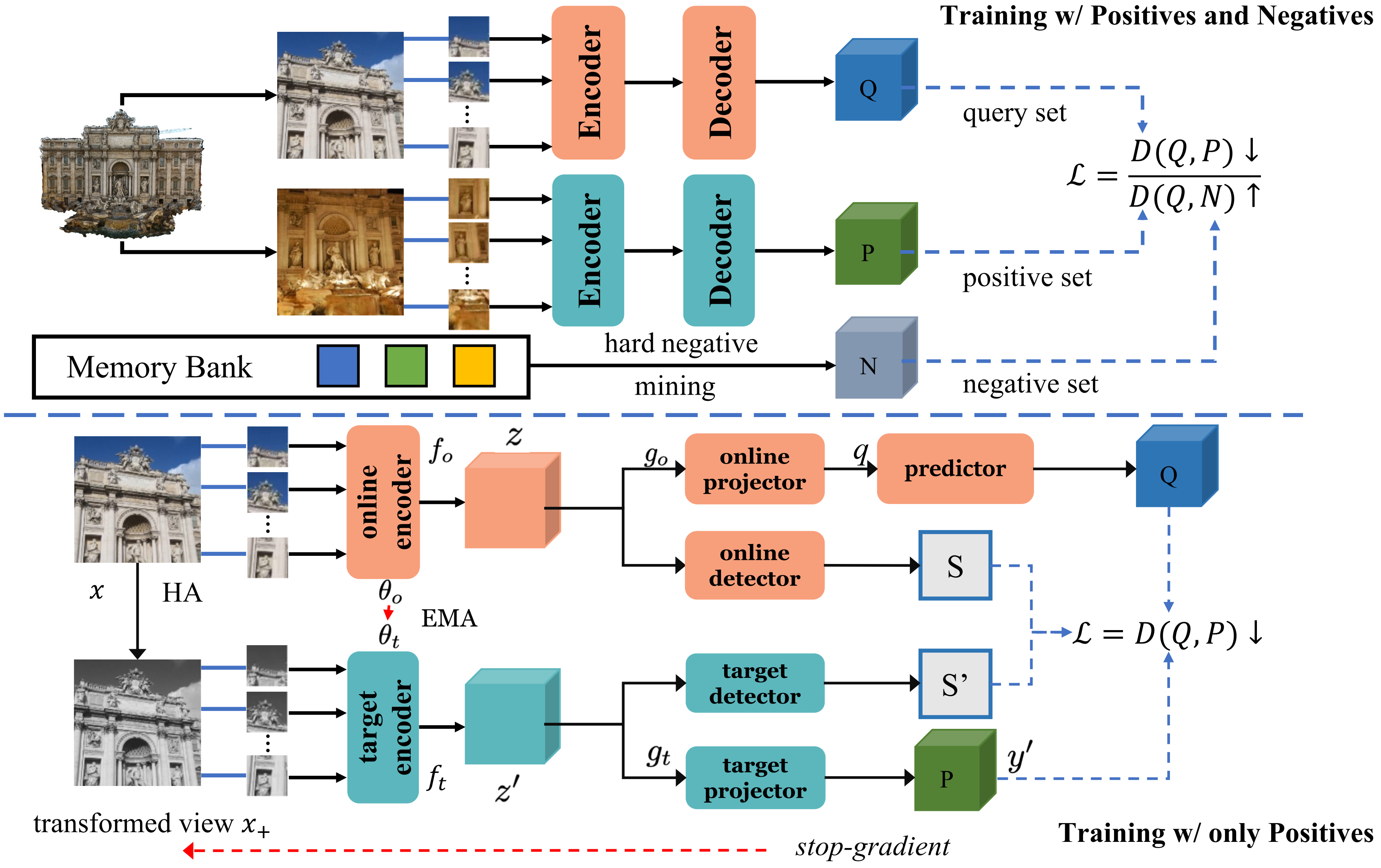
Motivation and Contribution



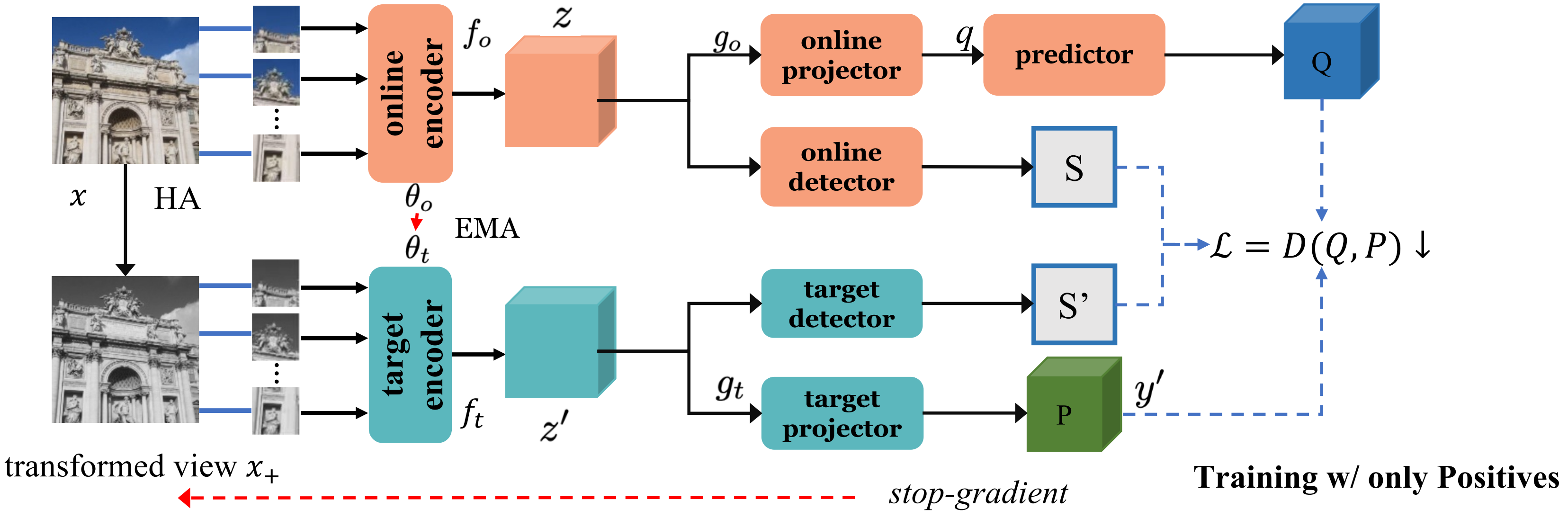
- **Motivation II**:
  - hard positives are encouraged as training data to expose novel patterns, while increasing the training difficulty.
  - All positives with different transformation strength are all labeled as coarse '*1*'.
- **Contribution II**:

  We adopt self-supervised generation learning and curriculum learning to soften the hard positives into continuous soft labels.

Overall Architecture



Method



Target encoder params are Exponential Moving Average of online encoder:

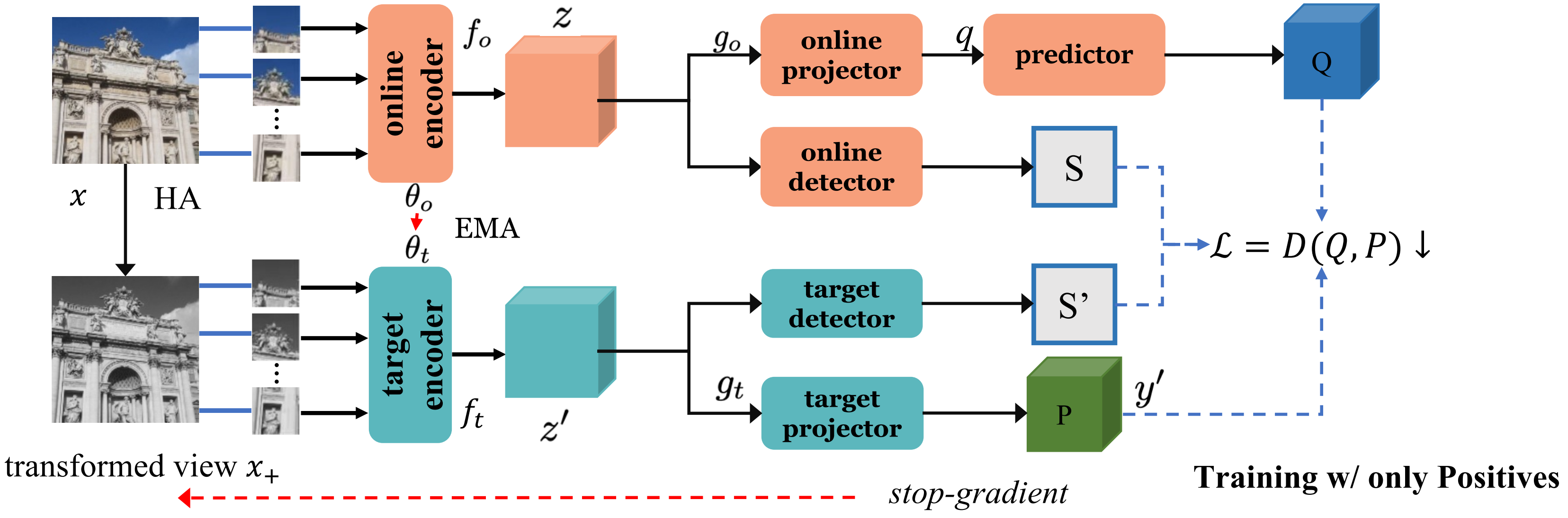
$$\theta_t \leftarrow \tau\theta_t + (1-\tau)\,\theta_o$$

Learning without Negative Samples!

The transformation prediction loss is computed on the corresponding locations:

$$\begin{aligned}\mathcal{L}_{\text{pred}}^{c} &= \left(1 - \langle y_c, y_c' \rangle\right) \\ &= \left(1 - \langle q\left(g_o\left(z\right)\right)_c, g_t\left(z'\right)_c \rangle\right)\end{aligned}$$

Method



Learning with Soft Labels!

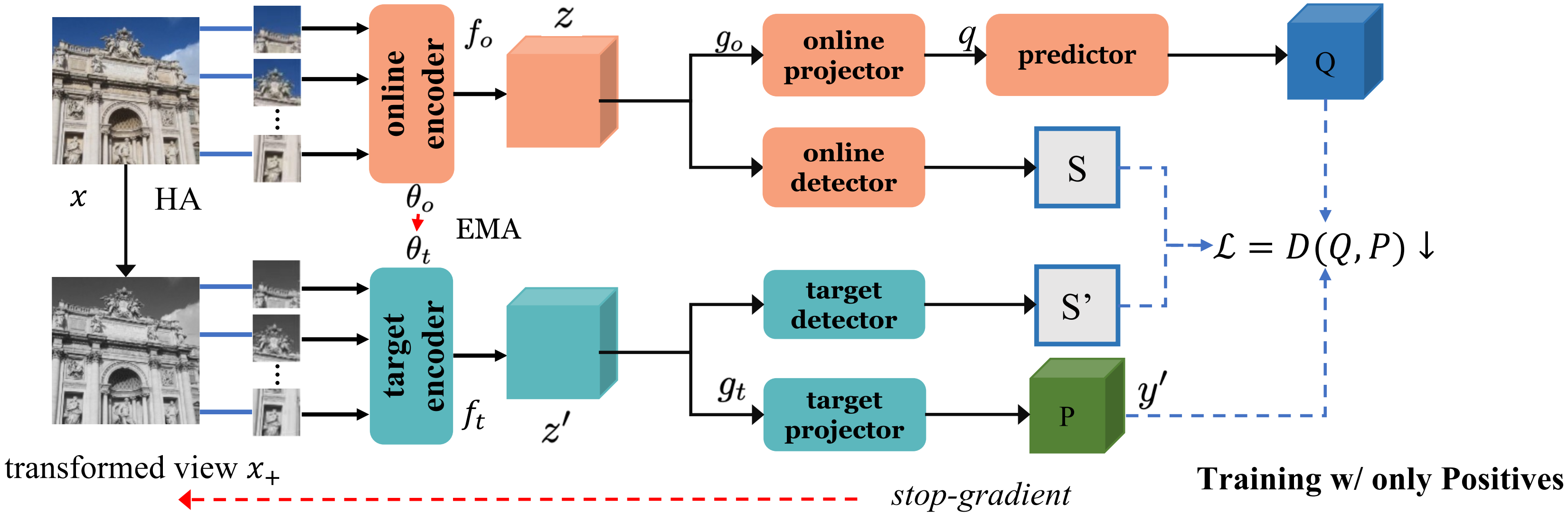
Contrastive loss with **hard** positive label, i.e., 1:

$$\mathcal{L}_{\text{hard}} = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} \frac{s_c s'_c}{\sum_{n\in\mathcal{C}} s_n s'_n} \mathcal{L}^c_{\text{pred}}$$

Contrastive loss with **soft** positive label:

$$\mathcal{L}_{\text{soft}} = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} \frac{s_c s'_c}{\sum_{n\in\mathcal{C}} s_n s'_n} \left(l_c + 1 - \mathcal{L}^c_{\text{pred}}\right)$$

Method



How to generate Soft Labels?

Self-supervised Generation Learning: $l_c = e^{-\left(1-\left\langle y_c^{\omega-1}, {y'}_c^{\omega-1}\right\rangle\right)/\lambda}$

Curriculum Setting for Positives Generation: $l_c = e^{-\alpha\left(1-\left\langle y_c^{\omega-1}, {y'}_c^{\omega-1}\right\rangle\right)/\lambda}$

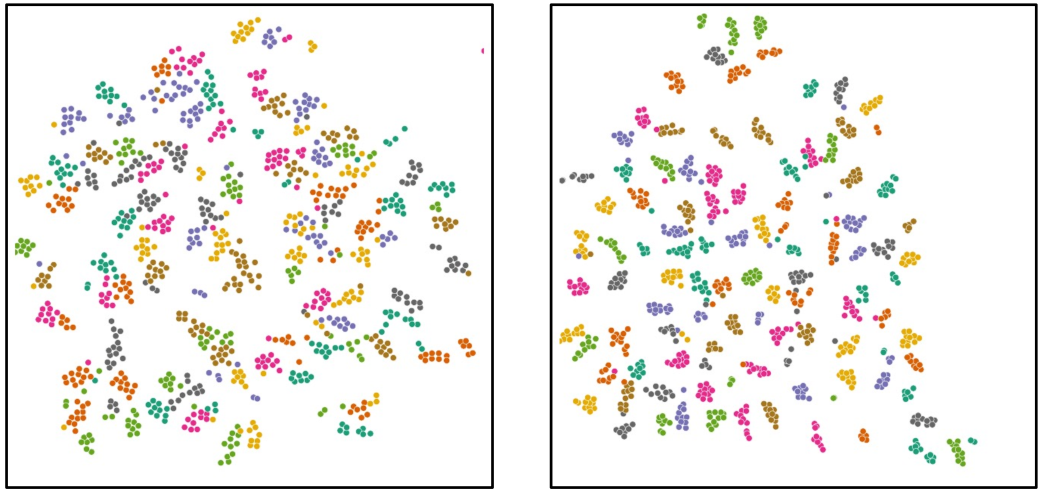
JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Experiments

Method	MMA@3	AUC@2	AUC@5
SIFT [28]	50.1	39.49	49.57
HardNet [33]	62.1	42.61	56.85
LF-Net [36]	53.2	38.74	48.69
SuperPoint [11]	65.7	44.08	59.04
DELF [35]	50.7	44.73	49.70
ContextDesc [29]	63.2	47.23	58.25
Key.Net [23]	72.1	40.87	56.04
R2D2 [40]	72.1	43.35	64.17
DISK [52]	77.2	52.33	69.80
ALIKE [63]	70.5	51.65	69.04
SSL+CAPS [31]	69.0	48.72	62.19
LLF [49]	74.0	52.14	66.81
MTLDesc [53]	78.7	55.02	71.42
PoSFeat [24]	75.34	50.16	69.23
D2-Net [12] (*orig.*)	40.3	19.49	37.78
D2-Net [12] (*our impl.*)	44.5	22.35	43.17
Ours(VGG)	**49.6** ↑ 9.3	**24.46** ↑ 4.97	**47.69** ↑ 9.91
ASLFeat [30] (*orig.*)	72.2	50.10	66.93
ASLFeat [30] (*our impl.*)	74.4	51.83	69.24
Ours(DCN)	**75.5** ↑ 3.2	**52.33** ↑ 2.23	**70.15** ↑ 3.22
Ours(TR)	**79.8**	**57.18**	**73.00**

Comparative results on Hpatches.

We train key-points based on different backbone with our training methods, including VGG (D2-Net), DCN (ASLFeat), and Swin Transformer.



t-SNE visualization of description from different training methods. Left: D2-Net, Right: Ours(VGG).

Experiments

Method	*Feat*	Accuracy @ Thresholds (%) ↑		
		0.25m,2°	0.5m,5°	5m,10°
RootSIFT [1]	11K	53.4	62.3	72.3
SuperPoint [11]	7K	68.1	85.9	94.8
D2-Net [12]	14K	67.0	86.4	97.4
R2D2 [40]	10K	70.7	85.3	96.9
ASLFeat [30]	10K	71.2	85.9	96.9
DISK [52]	10K	72.8	86.4	97.4
MTLDesc [53]	7K	74.3	86.9	96.9
Ours (TR)	10K	**74.3**	**89.0**	**98.4**

Performance on Aachen Day-Night Localization datasets

Method	FPS	RMSE/m ↓										
		00	01	02	04	05	06	07	08	09	10	*Avg.*
ORB [41]	**20.6**	59.46	610.35	72.68	19.26	238.60	83.46	72.72	66.06	119.21	63.52	140.53
SuperPoint [11]	6.5	162.78	**123.34**	13.52	1.06	6.36	**2.05**	12.15	8.66	8.20	5.10	34.32
D2-Net [12]	8.8	10.44	183.04	105.33	2.29	14.58	2.25	10.72	24.27	29.62	9.61	39.22
R2D2 [40]	7.8	49.62	515.96	60.14	3.90	123.05	62.44	53.84	62.54	73.30	43.32	104.81
SOSNet [51]	6.3	171.67	309.83	10.36	0.47	14.68	4.07	15.35	10.75	3.24	7.67	54.81
DISK [52]	6.5	32.77	149.98	18.67	0.45	5.97	4.38	12.88	32.85	4.33	4.81	26.71
Ours (TR)	6.2	**7.07**	164.39	**9.72**	**0.23**	**3.46**	2.12	**9.99**	**7.42**	**3.10**	**3.72**	**21.12**

Visual odometry localization performance based on different key-points in KITTI datasets.

Local descriptors trained with our method perform better on visual localization and odometry.

Thanks for Watching