JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

Unsupervised 3D Point Cloud Representation Learning by Triangle Constrained Contrast for Autonomous Driving

Bo Pang , Hongchi Xia , Cewu Lu

Shanghai Jiao Tong University

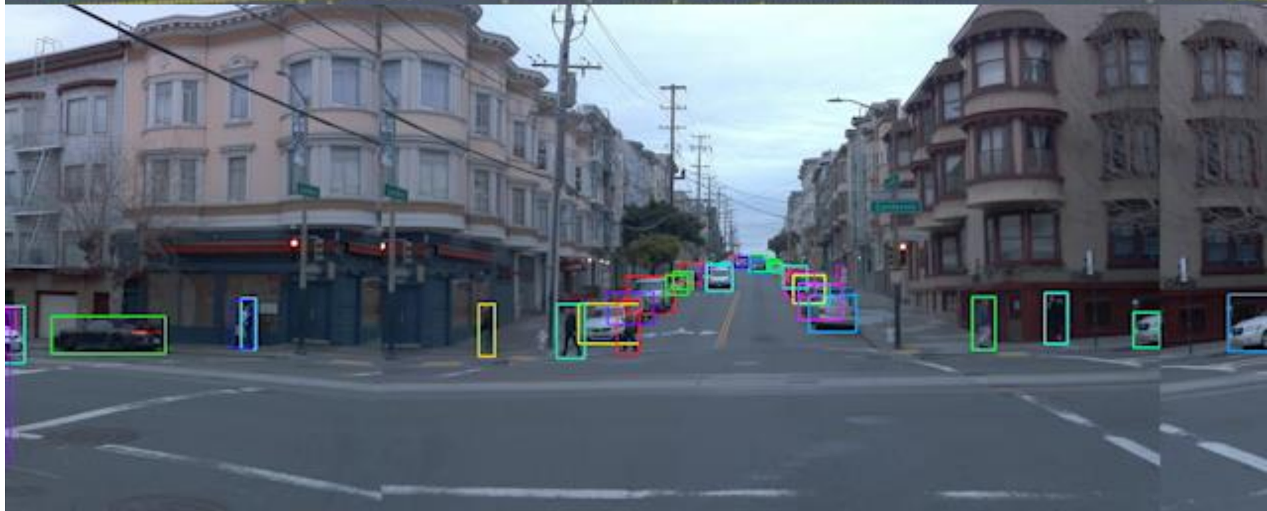
TUE-PM-104

Overview

- We aim to provide a unified unsupervised representation learning method that takes all the sensors of different modalities and both space-time dimensions into account.

- We design an automatic method to dig pixel/point-level multi-modality contrastive pairs across time.

- With the proposed TriCC pretraining method, we obtain effective 3D Lidar representations that perform SOTA on 3D segmentation and detection.



Overview

- We aim to provide a unified unsupervised representation learning method that takes all the sensors of different modalities and both space-time dimensions into account.
- We design an automatic method to dig pixel/point-level multi-modality contrastive pairs across time.
- With the proposed TriCC pretraining method, we obtain effective 3D Lidar representations that perform SOTA on 3D segmentation and detection.

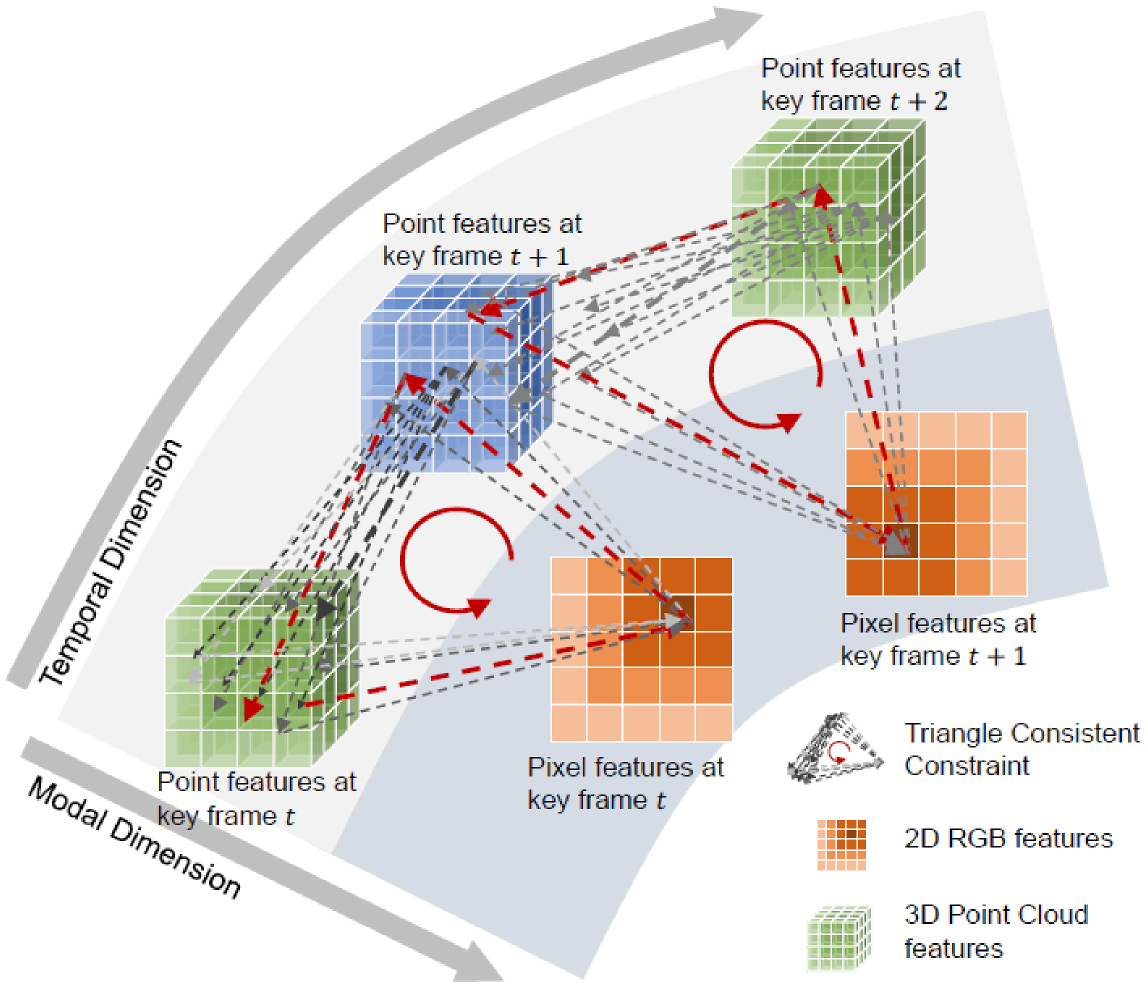
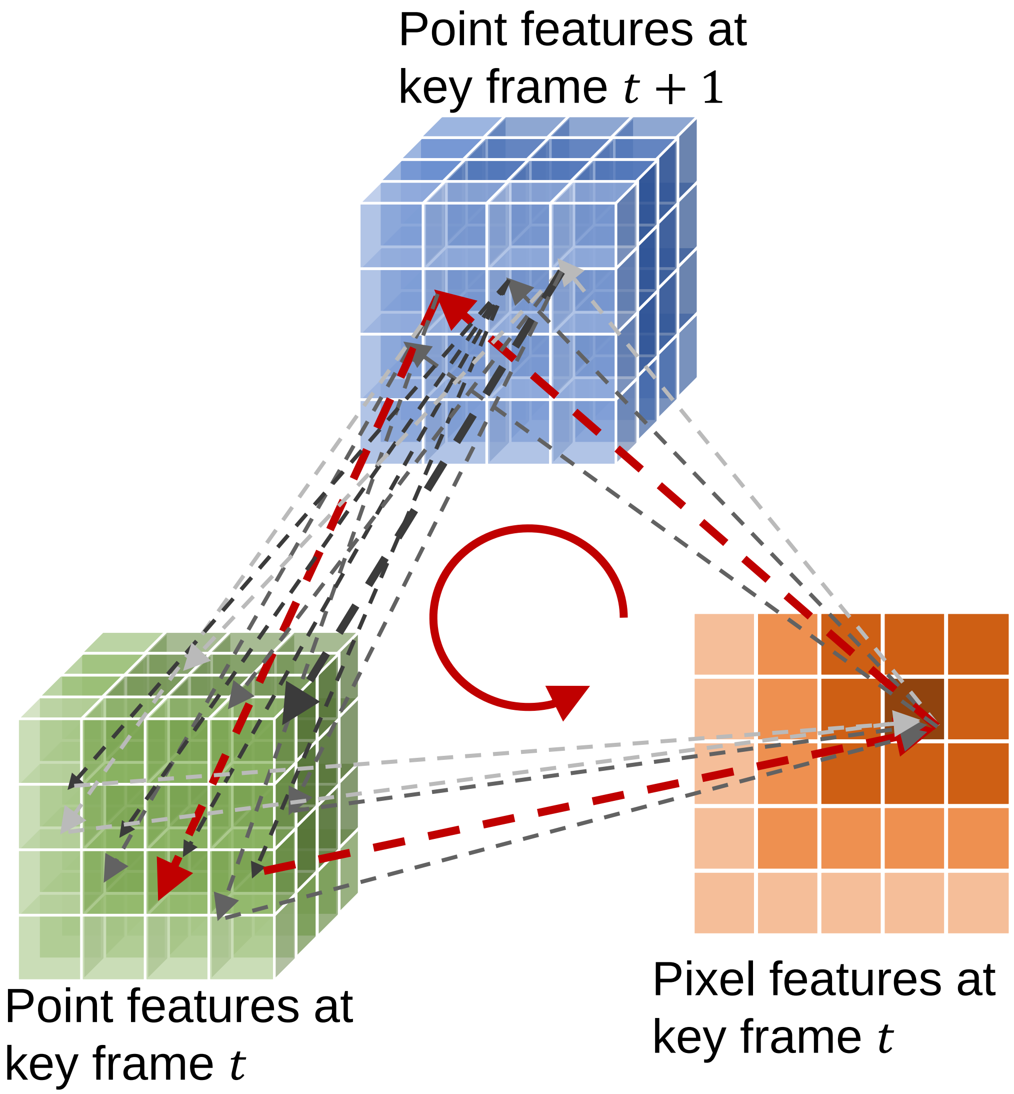


Illustration of Triangle consistent constraint

Overview

- We aim to provide a unified unsupervised representation learning method that takes all the sensors of different modalities and both space-time dimensions into account.
- We design an automatic method to dig pixel/point-level multi-modality contrastive pairs across time.
- With the proposed TriCC pretraining method, we obtain effective 3D Lidar representations that perform SOTA on 3D segmentation and detection.



Overview

- With the proposed TriCC pretraining method, we obtain effective 3D Lidar representations that perform SOTA on 3D segmentation and detection.

Pre-training Method	Pretrain Modality	Linear	Fine-tuning		
		100%	1%	10%	100%
Res16UNet as the backbone					
fine-tune from scratch	-	8.1	30.3	56.6	74.2
PointContrast [78]	P	21.9	32.5 (+2.2)	57.1 (+0.5)	74.3 (+0.1)
DepthContrast [88]	P	22.1	31.7 (+1.4)	57.3 (+0.7)	74.1 (-0.1)
PPKT [44]	P, C	36.4	37.8 (+7.5)	59.2 (+2.6)	73.8 (-0.4)
SLidR [59]	P, C	38.0	38.2 (+7.9)	58.8 (+2.2)	74.6 (+0.4)
TriCC(ours), 20 epoch	P, C	37.8	40.8 (+10.5)	60.2 (+3.6)	75.3 (+1.1)
TriCC(ours), 50 epoch	P, C	38.0	**41.2** (+10.9)	**60.4** (+3.8)	**75.6** (+1.4)
VoxelNet as the backbone					
fine-tune from scratch	-	2.6	24.5	43.1	53.9
SLidR [59]	P, C	33.5	32.1 (+7.6)	45.4 (+2.3)	54.3 (+0.4)
TriCC (ours), 20 epoch	P, C	33.6	**34.0** (+9.5)	**46.7** (+3.6)	**56.0** (+2.1)

Semantic segmentation on nuScenes

Pretrain	Det.	Fine-tuning		
		Easy	Moderate	Hard
mAP@R11 w/o road planes				
random	Sec.	73.3	63.2	60.3
SwAV [9]	Sec.	73.2 (-0.1)	64.0 (+0.8)	60.9 (+0.6)
DeepCluster [8]	Sec.	73.2 (-0.1)	63.4 (+0.2)	60.1 (-0.2)
STRL [34]	Sec.	74.0 (+0.7)	63.9 (+0.7)	60.9 (+0.6)
SLidR [59]	Sec.	73.6 (+0.3)	64.6 (+1.4)	61.5 (+1.2)
CO^3 [12]	Sec.	74.4 (+1.1)	64.4 (+1.2)	60.9 (+0.6)
TriCC (ours)	Sec.	**75.0** (+1.7)	**65.7** (+2.5)	**62.2** (+1.9)
mAP@R40 with road planes				
random	PV	81.3	70.6	66.1
Point Con. [78]	PV	82.8 (+1.5)	71.6 (+1.0)	67.5 (+1.4)
GCC-3D [41]	PV	-	71.3 (+0.7)	-
STRL [34]	PV	-	71.5 (+0.9)	-
SLidR [59]	PV	82.9 (+1.6)	71.9 (+1.3)	68.0 (+1.9)
Pro. Con. [84]	PV	**84.5** (+3.2)	72.9 (+2.3)	69.0 (+2.9)
TriCC (ours)	PV	84.1 (+2.8)	**73.3** (+2.7)	**69.4** (+3.3)

3D object detection results on KITTI

Motivation

1. Due to the difficulty of annotating the 3D LiDAR data of autonomous driving, an efficient unsupervised 3D representation learning method is important.

2. Currently there is no method can uniformly learn representations from all the information which can be accessed in auto-driving: camera and Lidar ones with the temporal dimension.

3. Dense positive pairs are difficult to obtain on multi-modality temporal information. We need an automatic pairing mechanism for many tasks.

Figure from nuscenes.org

Method Cycle consistent constraint

To get the dense positive pairs across multi-modality and temporal dimension, we design the Cycle Consistent Constraint to automatically dig them:

- Given a group of feature maps $\mathbf{X} = \{\mathbf{x}_i \in \mathbf{R}^{n_i \times c}, i = 1, \ldots, k\}$, we first define the transition matrix between two feature maps:

Similarity probability between two feature maps, we adopt cosine similarity in this work.

$$\mathbf{M} = \{\mathbf{m}_{i,j} = \mathrm{sm}(\langle \mathbf{x}_i, \mathbf{x}_j \rangle / \tau) \in \mathbb{R}^{n_i \times n_j}\}$$

- Then we need the transition matrices of adjacent features to form a cycle:

Accumulated multiplication of all the transition matrix

Connect the tail and head to form a cycle

$$\mathbf{S} \in \mathbb{R}^{n_1 \times n_1} = \left(\prod_{i=1}^{k-1} \mathbf{m}_{i,i+1}\right) \mathbf{m}_{k,1} = P(\mathbf{x_1} | \mathbf{x}_1)$$

Method Cycle consistent constraint

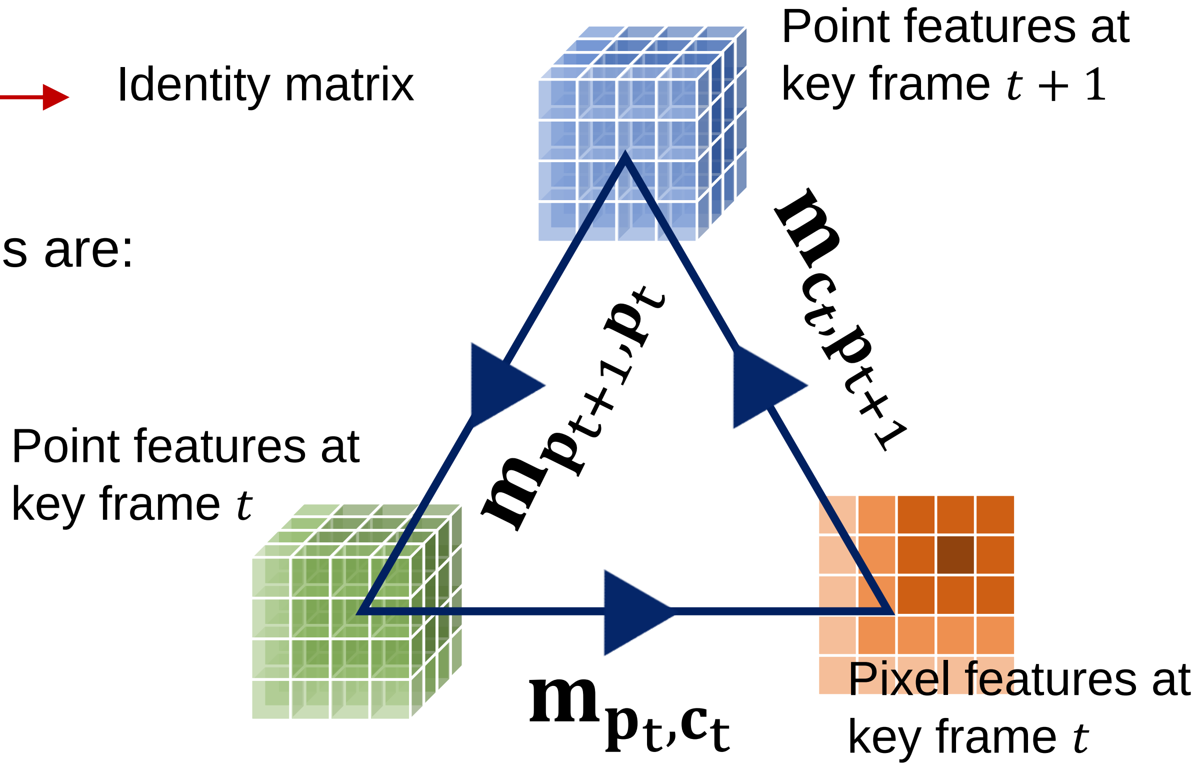
To get the dense positive pairs across multi-modality and temporal dimension, we design the Cycle Consistent Constraint to automatically dig them:

- With the cycle transition matrix **S**, the consistent constraint can be optimized by:

$$L_s = \text{CrossEntropy}(\log(\mathbf{S}), \underbrace{\mathbf{I}}_{\text{Identity matrix}})$$

- In our auto-driving scene, the feature maps are:

$$\mathbf{X} = \{\underbrace{\mathbf{C}_t}_{\text{Camera feature of frame } t}, \overbrace{\mathbf{P}_{t+1}}^{\text{Lidar feature of frame } t+1}, \underbrace{\mathbf{P}_t}_{\text{Lidar feature of frame } t}\}$$



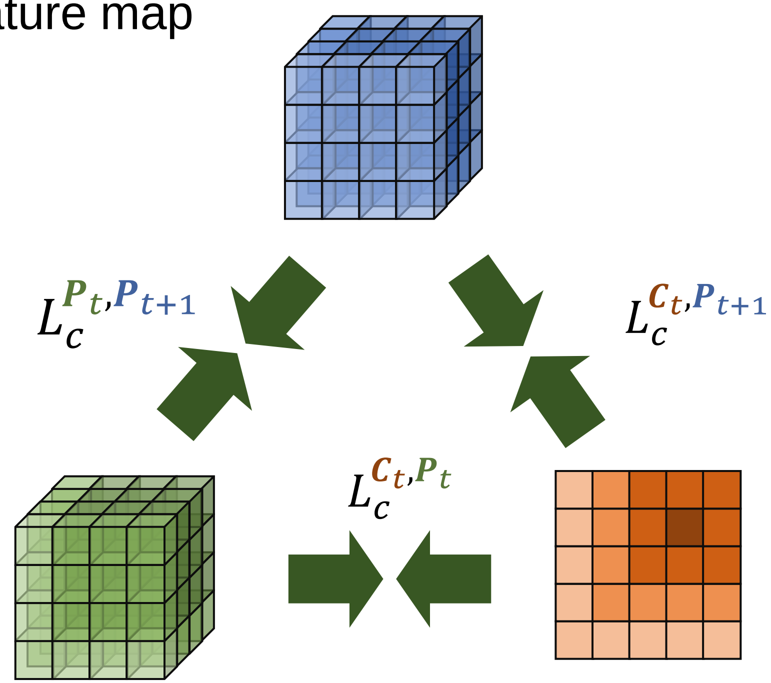
Method Triplet Contrast

With the automatically found positive pairs among the three feature maps, we can conduct the contrastive learning between each two of them.

- Take $\mathbf{C}_t$ and $\mathbf{P}_{t+1}$ as an example:

The qth feature vector in the feature map

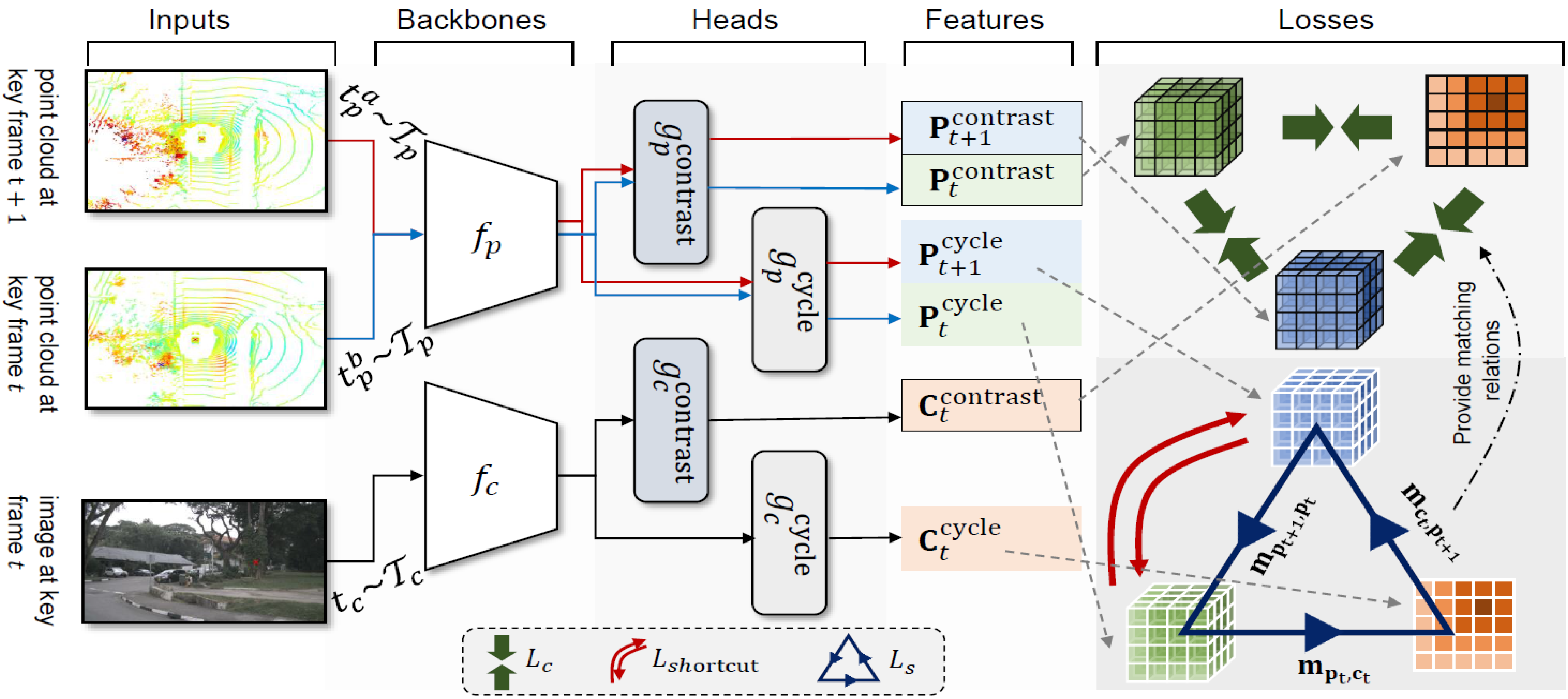
$$L_c^{\boldsymbol{C}_t,\boldsymbol{P}_{t+1}} = \frac{1}{n_{\boldsymbol{C}_t}} \sum_q -\log \frac{\exp(\mathrm{sim}(\boldsymbol{C}_t^q, \boldsymbol{P}_{t+1}^{\sigma\left(m^q_{\boldsymbol{C}_t,\boldsymbol{P}_{t+1}}\right)})/\tau)}{\sum_k \exp(\mathrm{sim}(\boldsymbol{C}_t^q, \boldsymbol{P}_{t+1}^k)/\tau)}$$



- The total contrastive loss is:

$$L_c = L_c^{\boldsymbol{C}_t,\boldsymbol{P}_{t+1}} + L_c^{\boldsymbol{C}_t,\boldsymbol{P}_t} + L_c^{\boldsymbol{P}_t,\boldsymbol{P}_{t+1}}$$

Method Pipeline



Experiments nuScenes semantic segmentation

Pre-training Method	Pretrain Modality	Linear 100%	Fine-tuning 1%	5%	10%	25%	100%
Res16UNet as the backbone							
fine-tune from scratch	-	8.1	30.3	47.7	56.6	64.8	74.2
PointContrast	P	21.9	32.5 (+2.2)	-	57.1 (+0.5)	-	74.3 (+0.1)
DepthContrast	P	22.1	31.7 (+1.4)	-	57.3 (+0.7)	-	74.1 (-0.1)
PPKT	P, C	36.4	37.8 (+7.5)	51.7 (+4.0)	59.2 (+2.6)	66.8 (+2.0)	73.8 (-0.4)
SLidR	P, C	38.0	38.2 (+7.9)	52.2 (+4.5)	58.8 (+2.2)	66.2 (+1.4)	74.6 (+0.4)
TriCC(ours), 20 epoch	P, C	37.8	40.8 (+10.5)	**54.1** (+6.4)	60.2 (+3.6)	**67.6** (+2.8)	75.3 (+1.1)
TriCC(ours), 50 epoch	P, C	38.0	**41.2** (+10.9)	**54.1** (+6.4)	**60.4** (+3.8)	**67.6** (+2.8)	**75.6** (+1.4)
VoxelNet as the backbone							
fine-tune from scratch	-	2.6	24.5	35.7	43.1	48.1	53.9
SLidR	P, C	33.5	32.1 (+7.6)	40.3 (+4.6)	45.4 (+2.3)	50.3 (+2.2)	54.3 (+0.4)
TriCC (ours), 20 epoch	P, C	33.6	**34.0** (+9.5)	**42.0** (+6.3)	**46.7** (+3.6)	**51.6** (+3.5)	**56.0** (+2.1)

Comparisons of different pre-training methods and different backbones under the linear probing and few-shots fine-tuning evaluation protocols on **nuScenes segmentation**.

Experiments

KITTI 3D Detection

We also transfer the pretrained representation on 3D detection task, it is seen that TriCC provides a 2.5 mAP and 2.7 mAP performance boost over the random initialization for SECOND and PV-RCNN models.

Pretrain	Det.	Fine-tuning		
		Easy	Moderate	Hard
mAP@R11 w/o road planes				
random	Sec.	73.3	63.2	60.3
SwAV	Sec.	73.2 (-0.1)	64.0 (+0.8)	60.9 (+0.6)
DeepCluster	Sec.	73.2 (-0.1)	63.4 (+0.2)	60.1 (-0.2)
BYOL	Sec.	71.1 (-2.2)	60.4 (-2.8)	57.0 (-3.3)
Point Con.	Sec.	72.7 (-0.6)	62.7 (-0.5)	59.2 (-1.1)
GCC-3D	Sec.	73.9 (+0.6)	63.5 (+0.3)	59.8 (-0.5)
STRL	Sec.	74.0 (+0.7)	63.9 (+0.7)	60.9 (+0.6)
SLidR	Sec.	73.6 (+0.3)	64.6 (+1.4)	61.5 (+1.2)
CO^3	Sec.	74.4 (+1.1)	64.4 (+1.2)	60.9 (+0.6)
TriCC (ours)	Sec.	**75.0** (+1.7)	**65.7** (+2.5)	**62.2** (+1.9)
mAP@R40 with road planes				
random	PV	81.3	70.6	66.1
Point Con.	PV	82.8 (+1.5)	71.6 (+1.0)	67.5 (+1.4)
GCC-3D	PV	-	71.3 (+0.7)	-
STRL	PV	-	71.5 (+0.9)	-
SLidR	PV	82.9 (+1.6)	71.9 (+1.3)	68.0 (+1.9)
Pro. Con.	PV	**84.5** (+3.2)	72.9 (+2.3)	69.0 (+2.9)
TriCC (ours)	PV	84.1 (+2.8)	**73.3** (+2.7)	**69.4** (+3.3)

Comparisons with SOTA 3D representation learning methods on KITTI fine-tuning with 100% annotations

Experiments More results

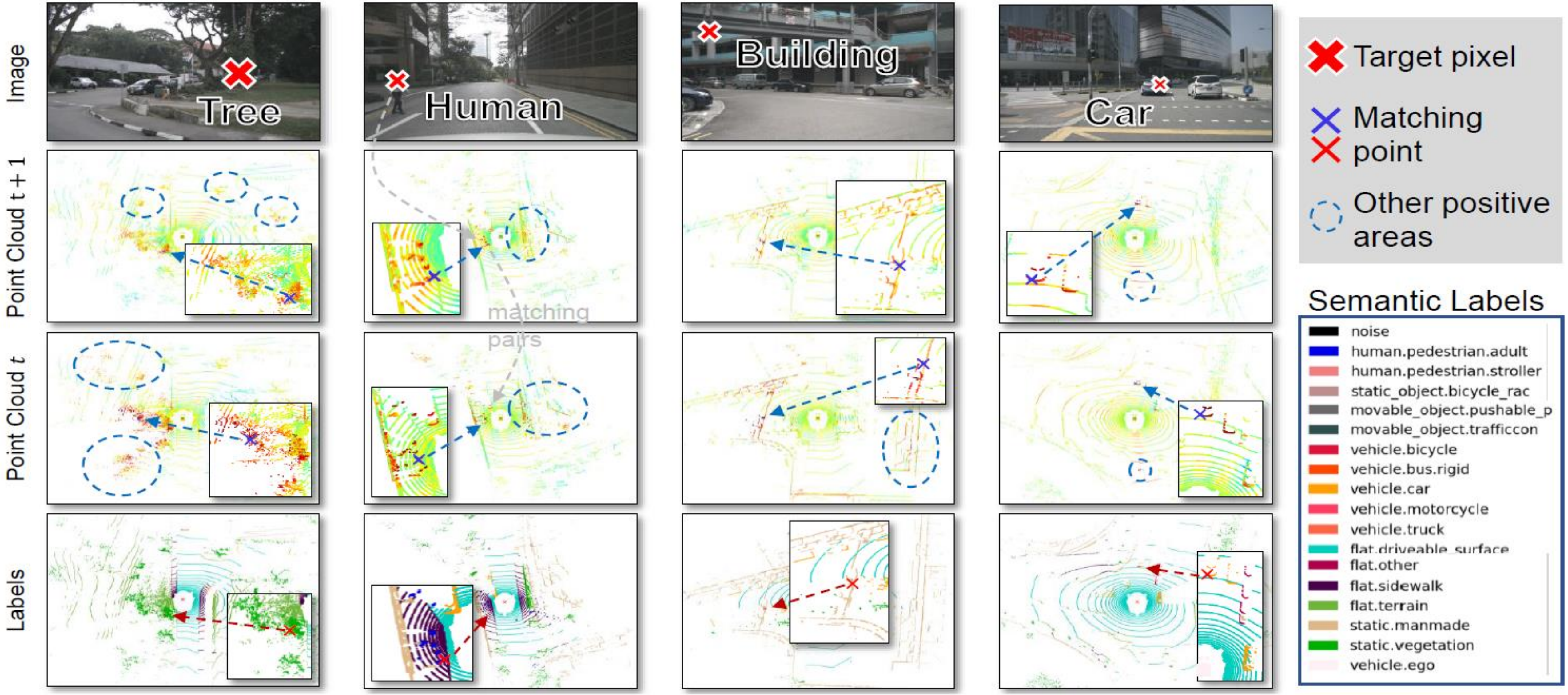
Pre-training Method	Fine-tuning 1%	5%	10%
Res16UNet as the backbone			
fine-tune from scratch	39.5	52.1	55.6
PPKT [44]	43.9 (+4.4)	53.1 (+1.0)	57.3 (+1.7)
SLidR [59]	44.6 (+5.1)	52.6 (+0.5)	56.0 (+0.4)
TriCC (ours), 20 epoch	45.8 (+6.3)	55.7 (+3.6)	58.4 (+2.8)
TriCC (ours), 50 epoch	**45.9** (+6.4)	**55.9** (+3.8)	**59.0** (+3.4)
VoxelNet as the backbone			
fine-tune from scratch	28.8	40.8	46.4
SLidR [59]	35.2 (+6.4)	45.5 (+4.7)	48.6 (+2.2)
TriCC (ours), 20 epoch	**36.5** (+7.7)	**46.8** (+6.0)	**49.8** (+3.4)

Few-shots fine-tuning segmentation results on SemanticKITTI

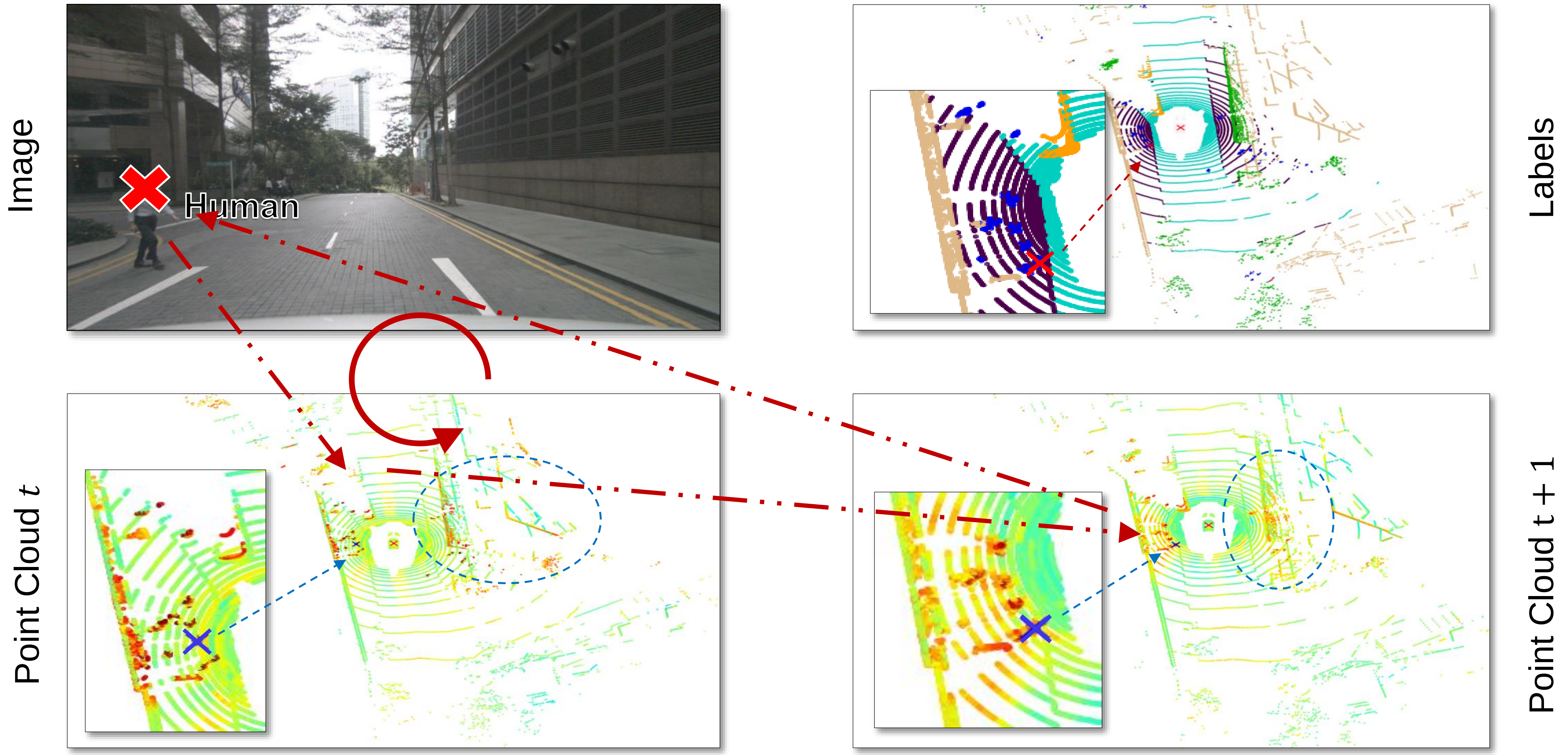
Pretrain	Fine-tuning 5% label		10% label		20% label	
	mAP	NDS	mAP	NDS	mAP	NDS
VoxelNet + CenterPoint						
random	38.0	44.3	46.9	55.5	50.2	59.7
Point Con. [78]	39.8	45.1	47.7	56.0	-	-
GCC-3D [41]	41.1	46.8	48.4	56.7	-	-
SLidR [59]	43.3	52.4	47.5	56.8	50.4	59.9
TriCC, 20epoch	**44.6**	**54.4**	**48.9**	**58.1**	**50.9**	**60.3**
VoxelNet + SECOND						
random	35.8	45.9	39.0	51.2	43.1	55.7
SLidR [59]	36.6	48.1	39.8	52.1	44.2	56.3
TriCC, 20epoch	**37.8**	**50.0**	**41.4**	**53.5**	**45.5**	**57.7**

Few-shots fine-tuning 3D detection results on nuScenes

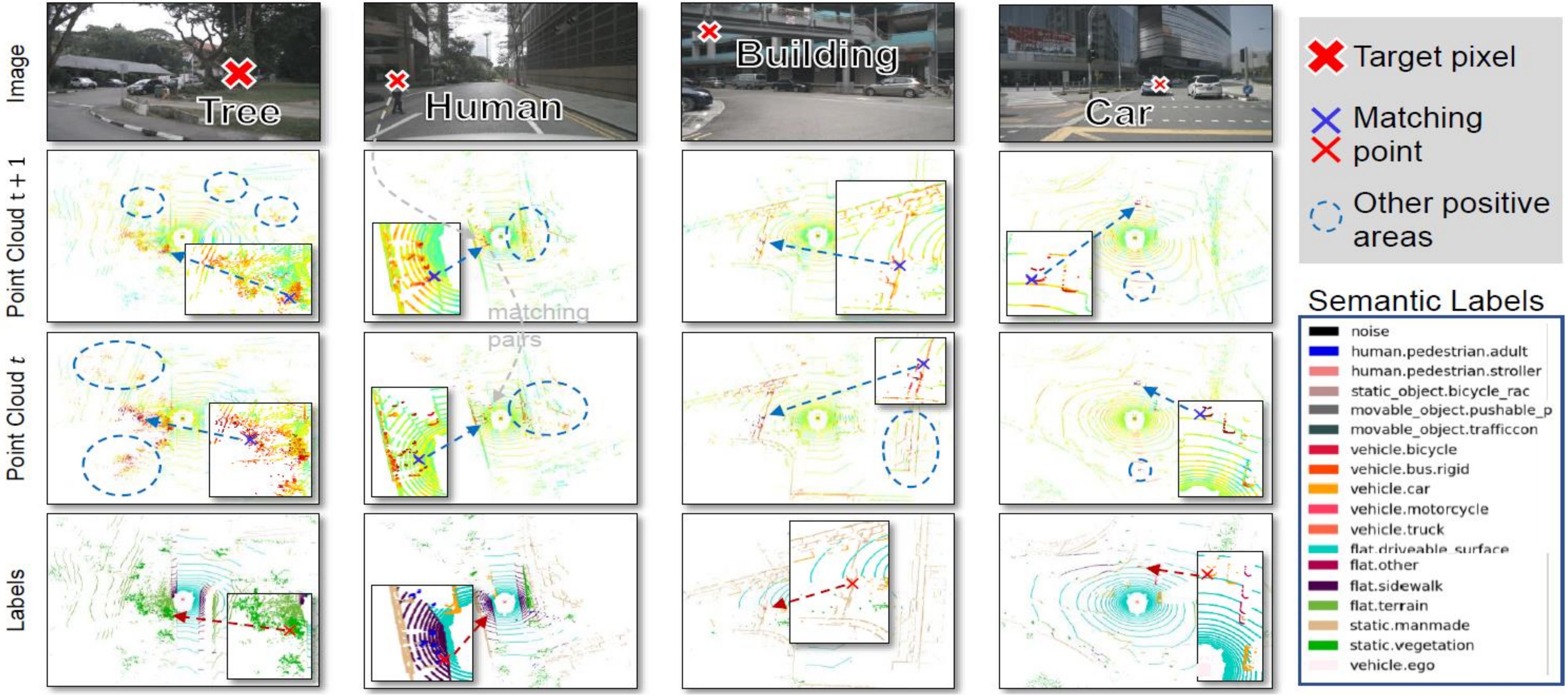
Experiments Vis analysis



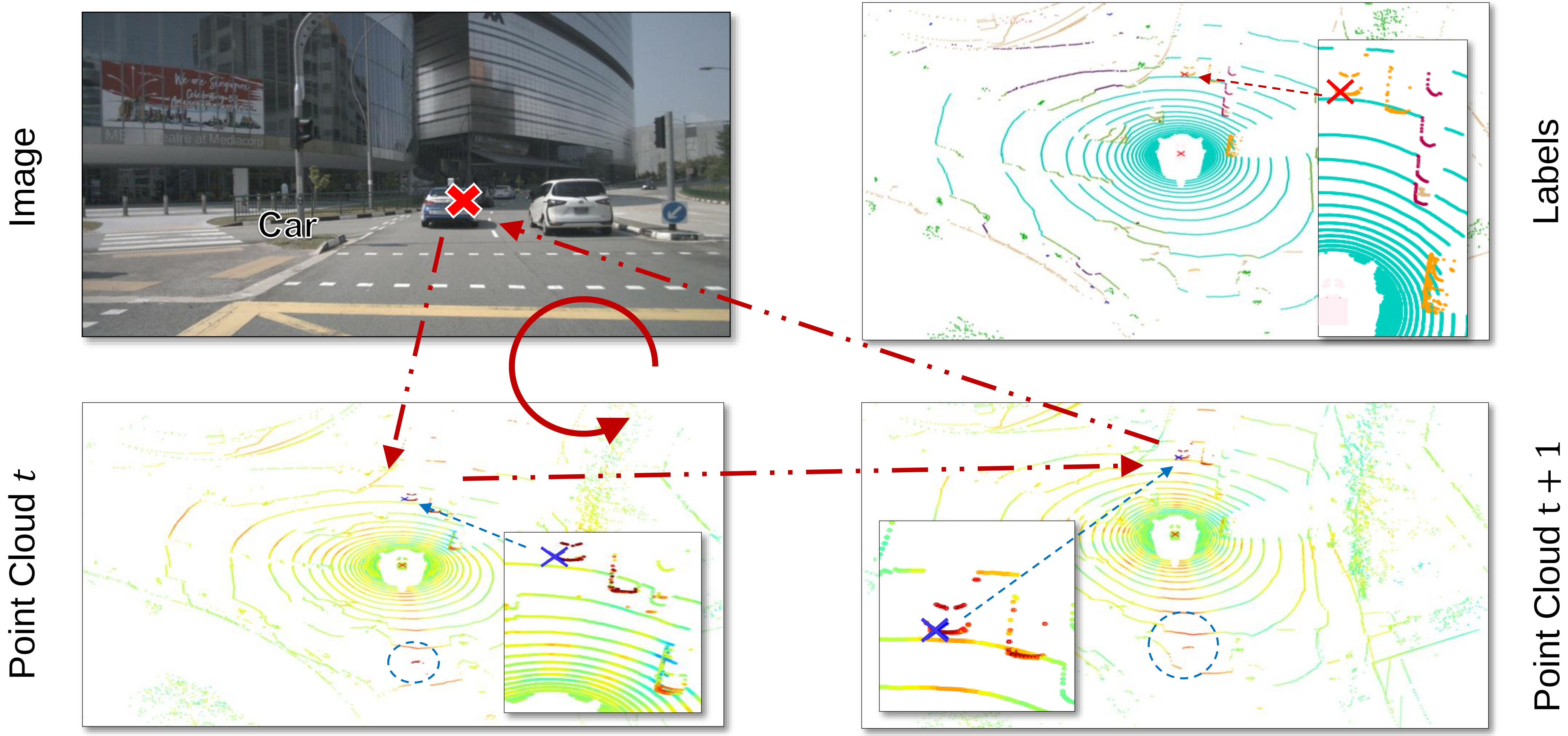
Experiments Vis analysis



Experiments Vis analysis



Experiments Vis analysis



JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Thanks