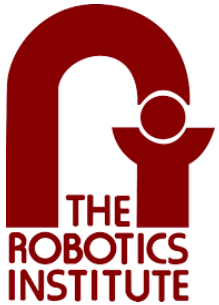
Self-supervised Learning for Autonomous Driving

David Held

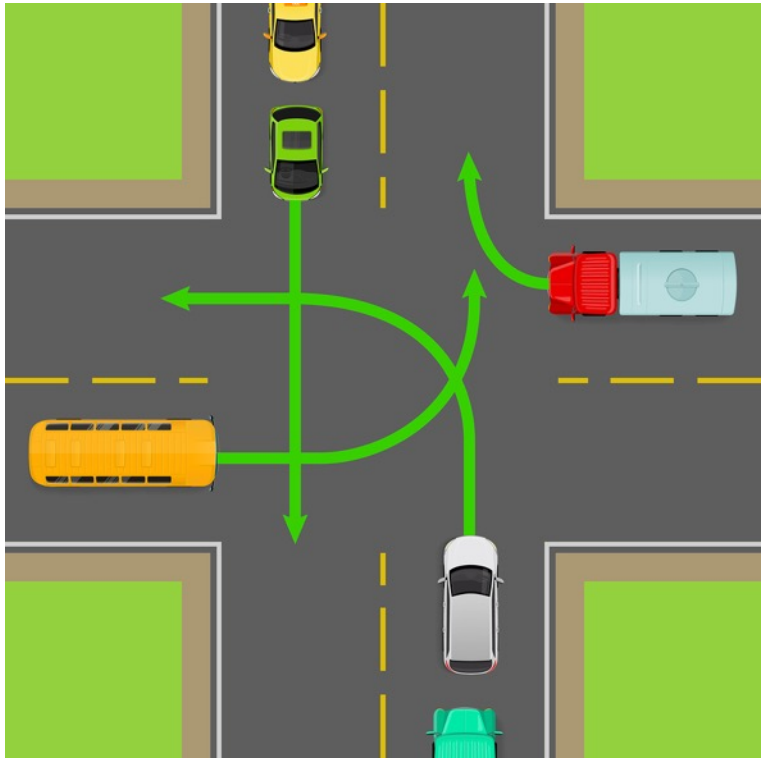
Carnegie Mellon University



Why is autonomous driving so hard?

[https://driving-tests.org/beginner-drivers/how-to-drive-on-the-highway/]

Why is autonomous driving so hard?



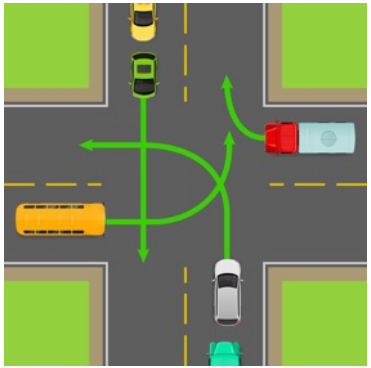
[https://www.dolmanlaw.com/causes-intersection-crashes/]

What about this:

[https://www.youtube.com/watch?v=Ucp0TTmvqOE&feature=youtu.be&t=8659]

[https://www.braincreators.com/brainpower/insights/teslas-data-engine-and-what-we-should-all-learn-from-it]

The Long Tail of Autonomous Driving



Regular, easy driving

Difficult driving

Weird driving

https://en.wikipedia.org/wiki/Long_tail#/media/File:Long_tail.svg

Picture by Hay Kranen / PD

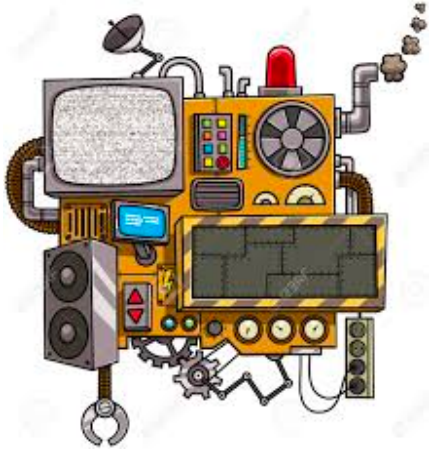
[https://www.youtube.com/watch?v=Ucp0TTmvqOE&feature=youtu.be&t=8659]

[https://www.braincreators.com/brainpower/insights/teslas-data-engine-and-what-we-should-all-learn-from-it]

Traditional ML: Supervised Learning

Large image dataset

Human labeling



Machine Learning

System Output

Problem: How to ensure that we have labeled all of the weird cases?

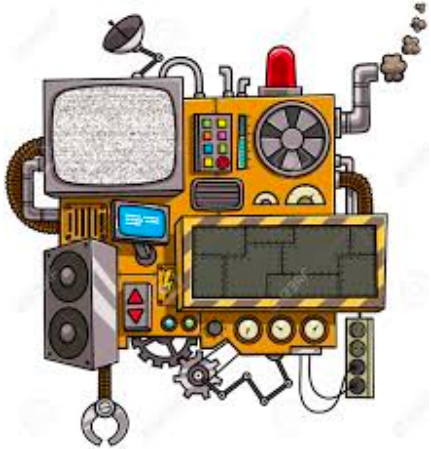
Large image dataset

Human labeling

Traditional ML: Supervised Learning

Large image dataset

Human labeling

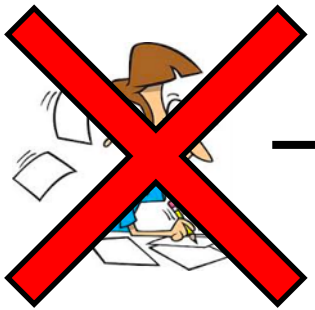


Machine Learning

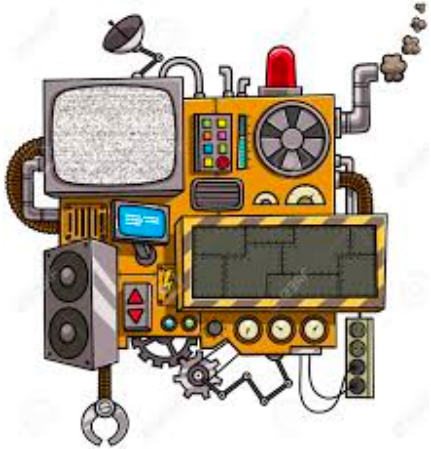
System Output

Self-Supervised Learning

Large image dataset



Human labeling



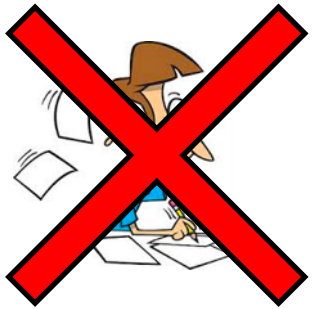
Machine Learning

DOG, DOG, CAT

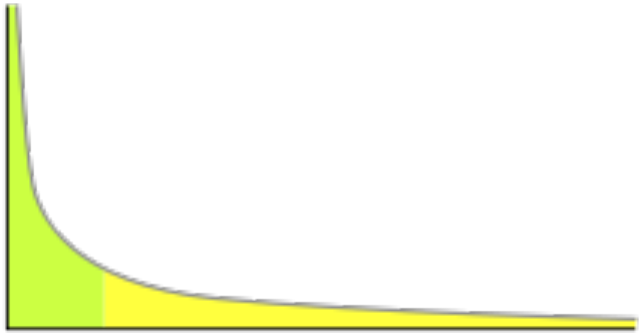
System Output

Self-Supervised Learning for Autonomous Driving

- Active Perception with Light Curtains (ECCV 2020)
- Self-supervised 3D Scene Flow (CVPR 2020)
- Self-supervised 3D Data Association (IROS 2020)



Human labeling



Regular, easy driving

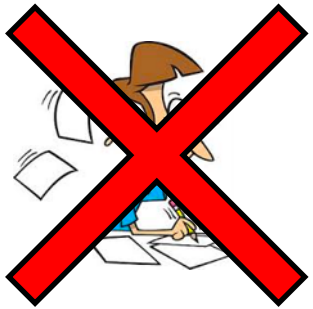
Difficult driving

Weird driving

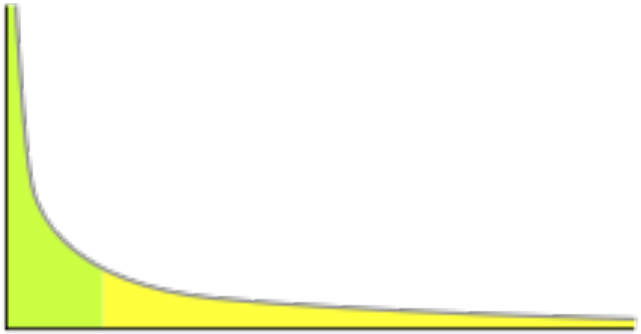
Self-Supervised Learning for Autonomous Driving

- **Active Perception with Light Curtains (ECCV 2020)**
- Self-supervised 3D Scene Flow (CVPR 2020)
- Self-supervised 3D Data Association (IROS 2020)

Sid Ancha



Human labeling

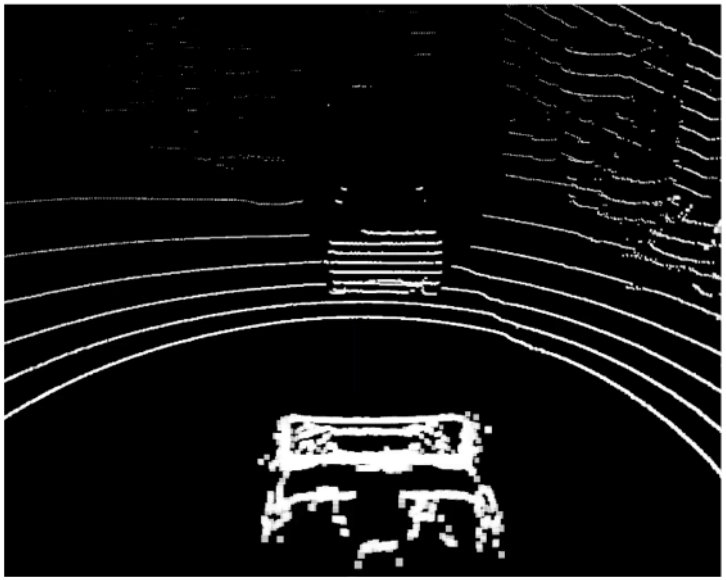


Regular, easy driving

Difficult driving

Weird driving

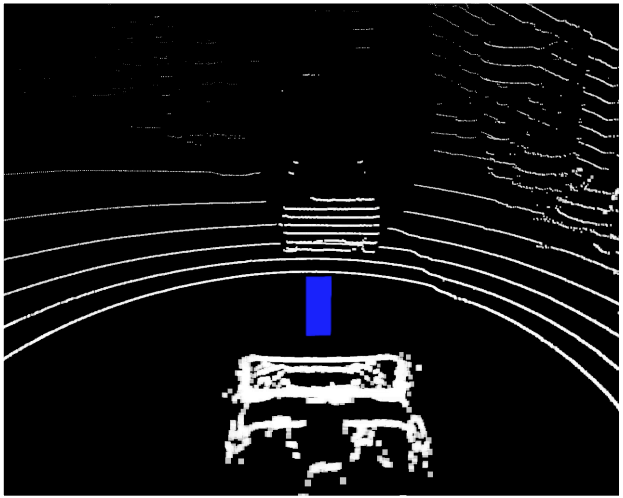
LiDARs perform fixed scans

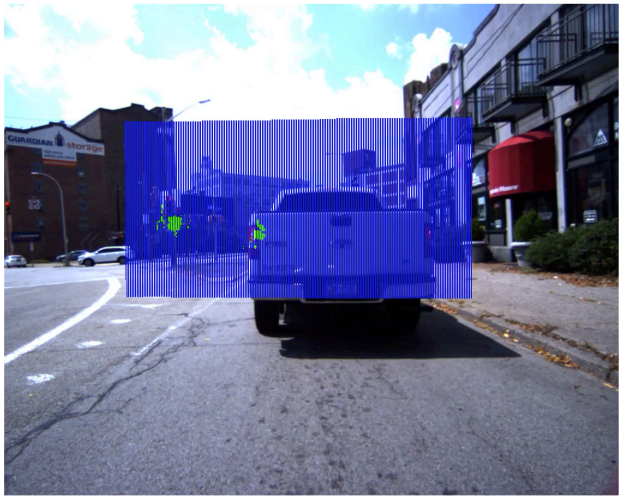


LiDAR

- **Sparse point clouds.**
- **Expensive: Velodyne 64 beam LiDAR can cost > $80,000.**

Light curtains are *controllable*





LiDAR

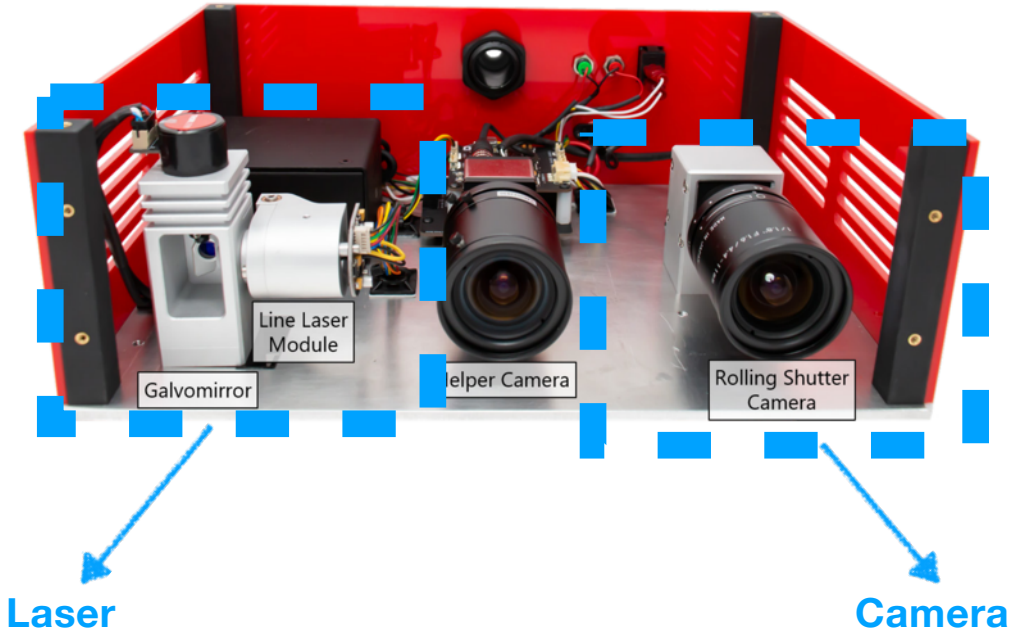
- Sparse point clouds.
- Expensive: Velodyne 64 beam LiDAR can cost > $80,000.

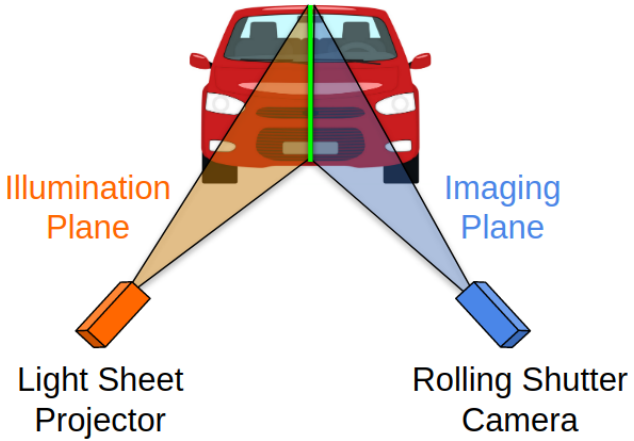
Light Curtain

- Dense point cloud where curtain is placed.
- Inexpensive:
  Lab-built prototype costs ~$1000.

**Agile Depth Sensing Using Triangulation Light Curtains,* Bartels et. al. 2019

What is a light curtain?

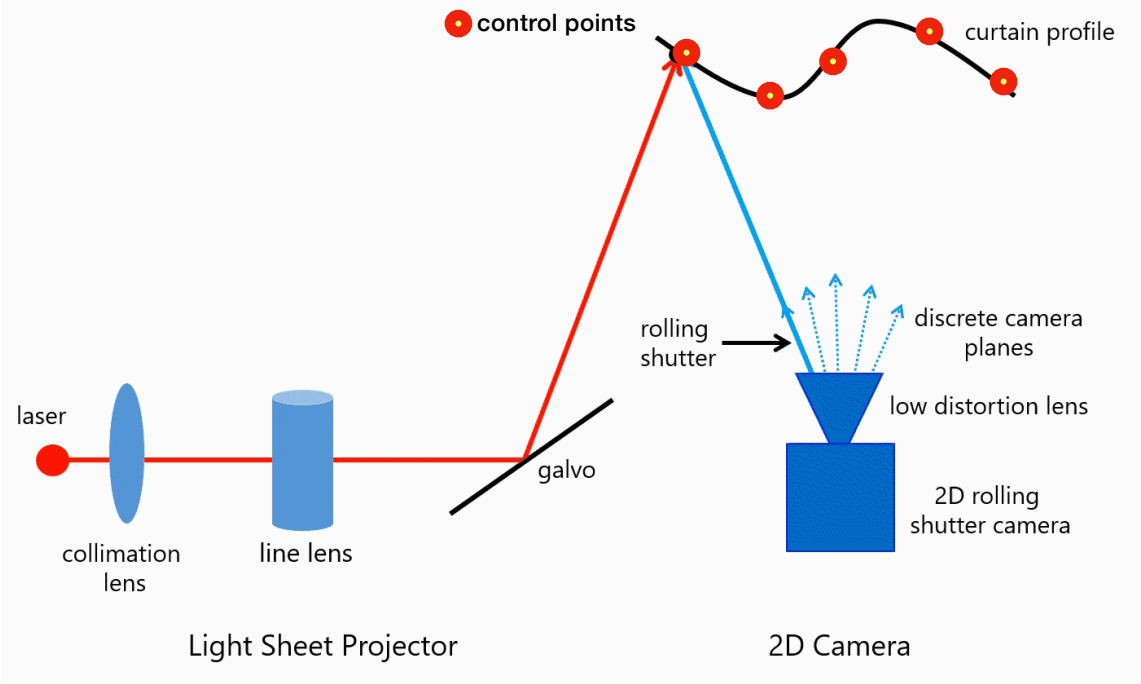




"Agile Depth Sensing Using Triangulation Light Curtains", Bartels et. al.

`http://www.cs.cmu.edu/~ILIM/agile_depth_sensing`

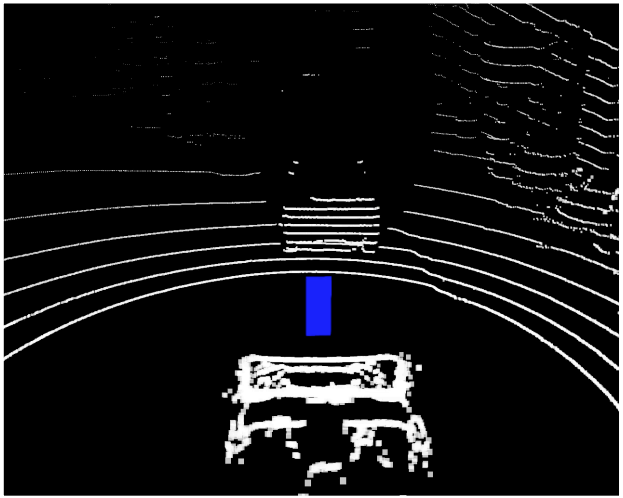
Light curtain working principle



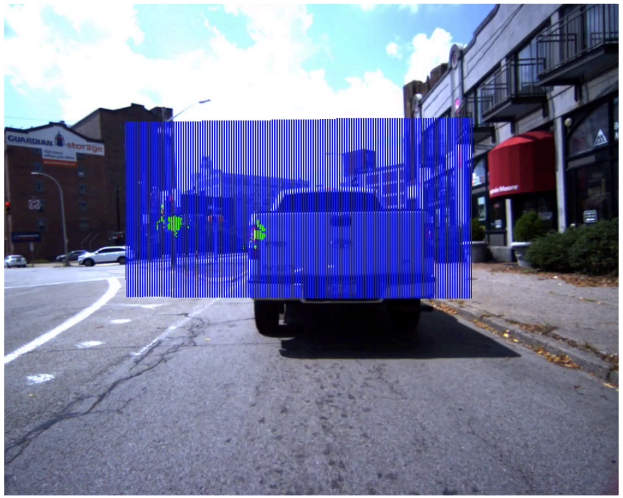
"Agile Depth Sensing Using Triangulation Light Curtains", Bartels et. al.

http://www.cs.cmu.edu/~ILIM/agile_depth_sensing

Light curtains output



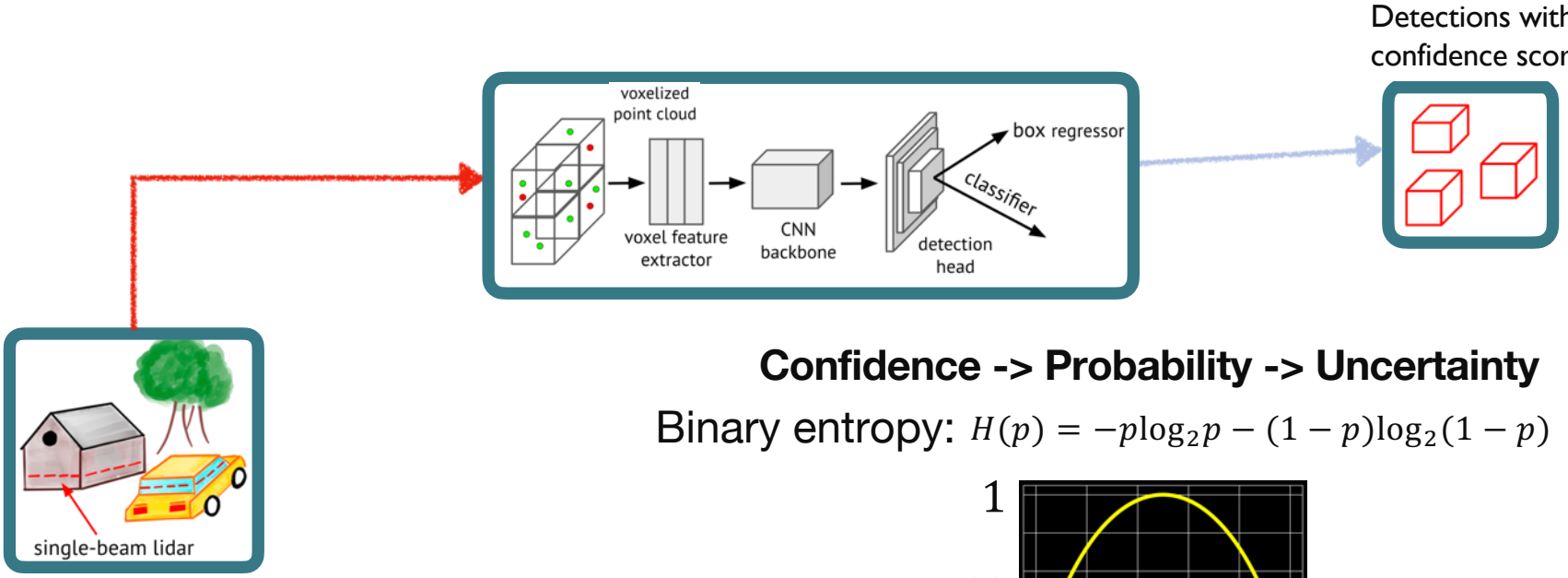
LiDAR



Light Curtain

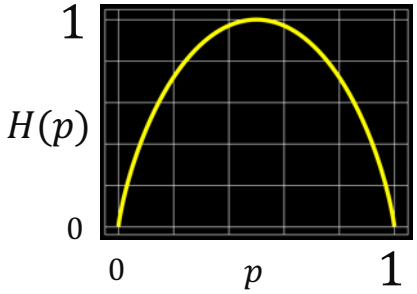
Agile Depth Sensing Using Triangulation Light Curtains,* **Bartels et. al. 2019

Where to Place Light Curtains?

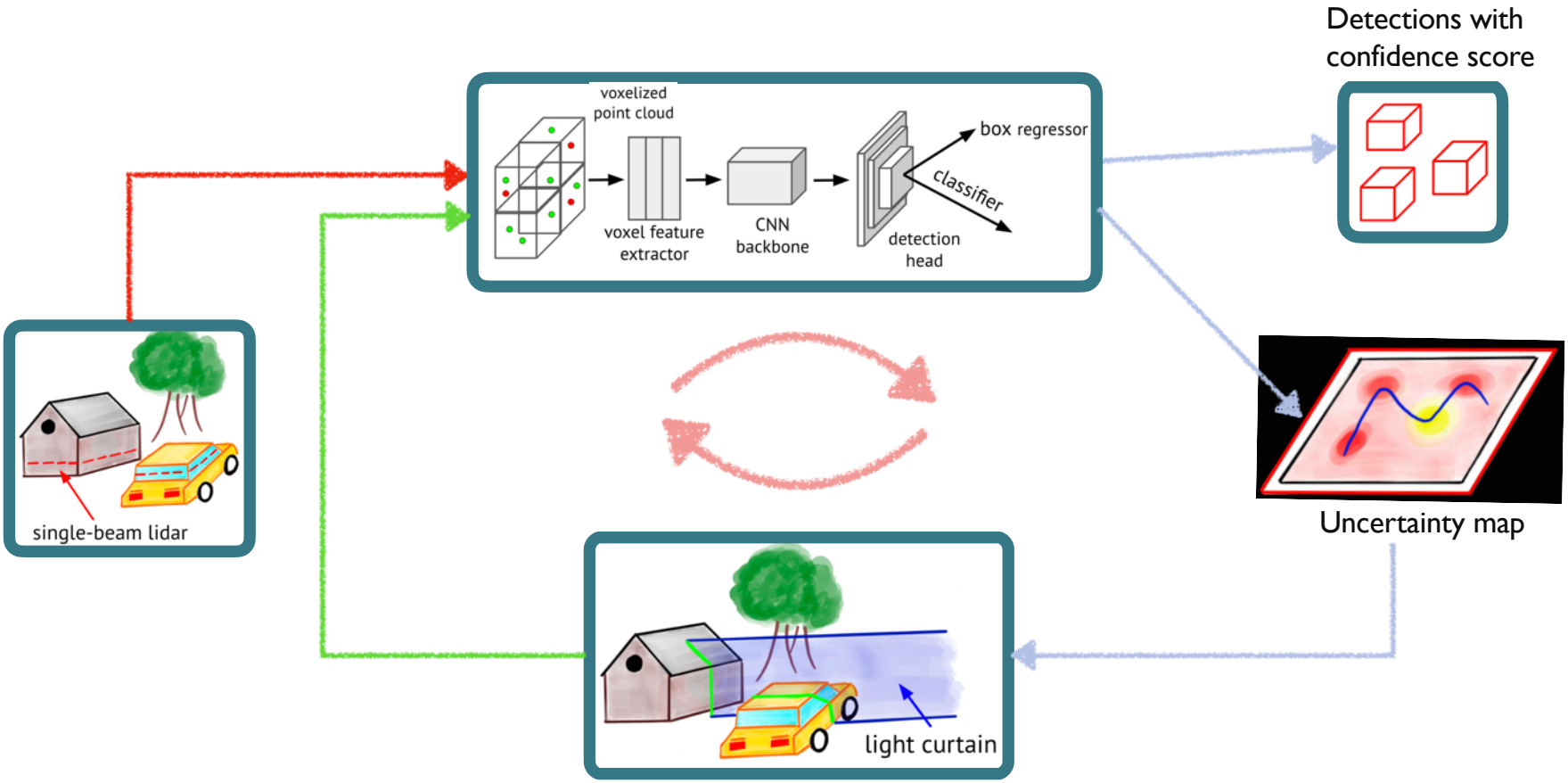


Confidence -> Probability -> Uncertainty

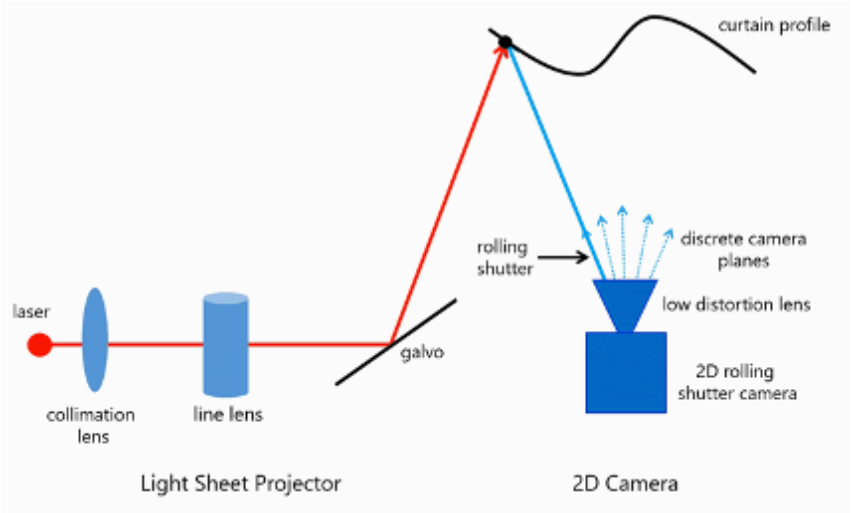
Binary entropy: $H(p) = -p\log_2 p - (1-p)\log_2(1-p)$



Where to Place Light Curtains?



Light curtain constraints



$$\theta_{\text{las}} \leq \omega_{\text{max}} \cdot \Delta t$$

Galvo max velocity limit

Time between consecutive rays (given by camera shutter speed)

"Agile Depth Sensing Using Triangulation Light Curtains", Bartels et. al.

http://www.cs.cmu.edu/~ILIM/agile_depth_sensing

Light curtain optimization

Place light curtain according to:

Maximize information gain

$$= \sum_k H(p_k^t) - \sum_k H(p_k^{t-1})$$

Subject to light curtain constraints

$$\theta_{\text{las}} \leq \omega_{\max} \cdot \Delta t$$

Light curtain optimization

Place light curtain according to:

Maximize information gain

$$= \sum_k H(p_k^t) - \sum_k H(p_k^{t-1})$$

Subject to light curtain constraints

$$\theta_{\text{las}} \leq \omega_{\max} \cdot \Delta t$$

Place light curtain according to:

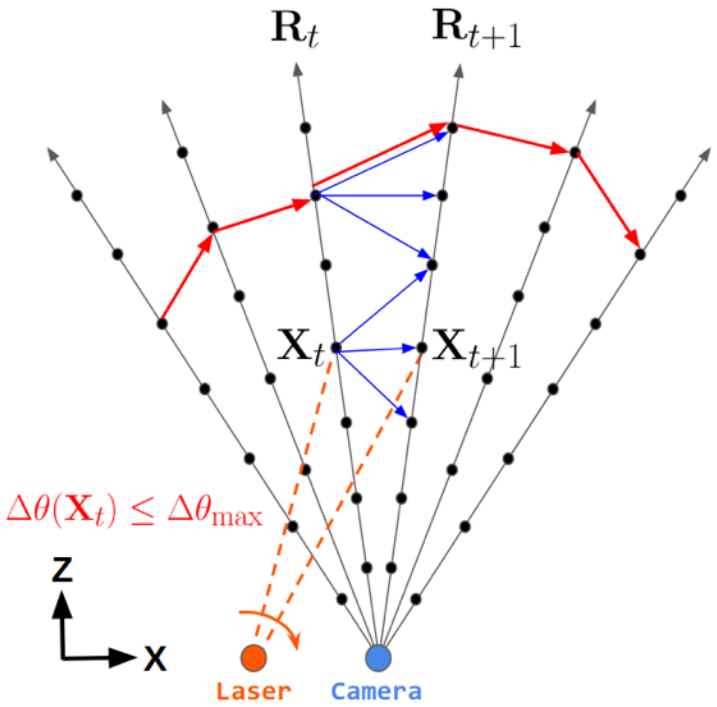
Maximize information gain

$$=\sum_k H(p_k^t) - \sum_k H(p_k^{t-1})$$

Subject to light curtain constraints

$$\theta_{\text{las}} \leq \omega_{\max} \cdot \Delta t$$

Constraint Graph



Place light curtain according to:

Maximize information gain

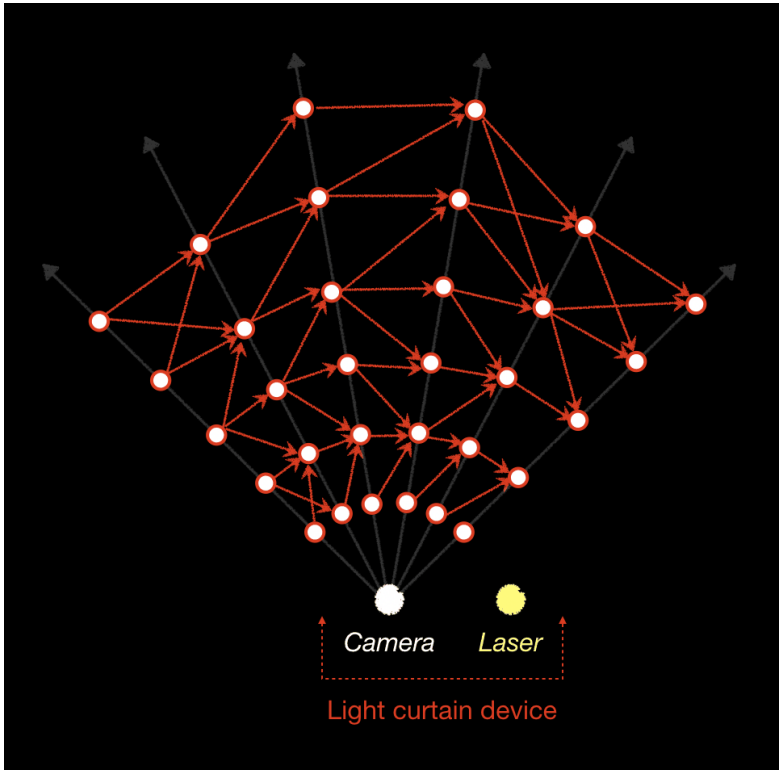
$$=\sum_k H(p_k^t) - \sum_k H(p_k^{t-1})$$

Subject to light curtain constraints

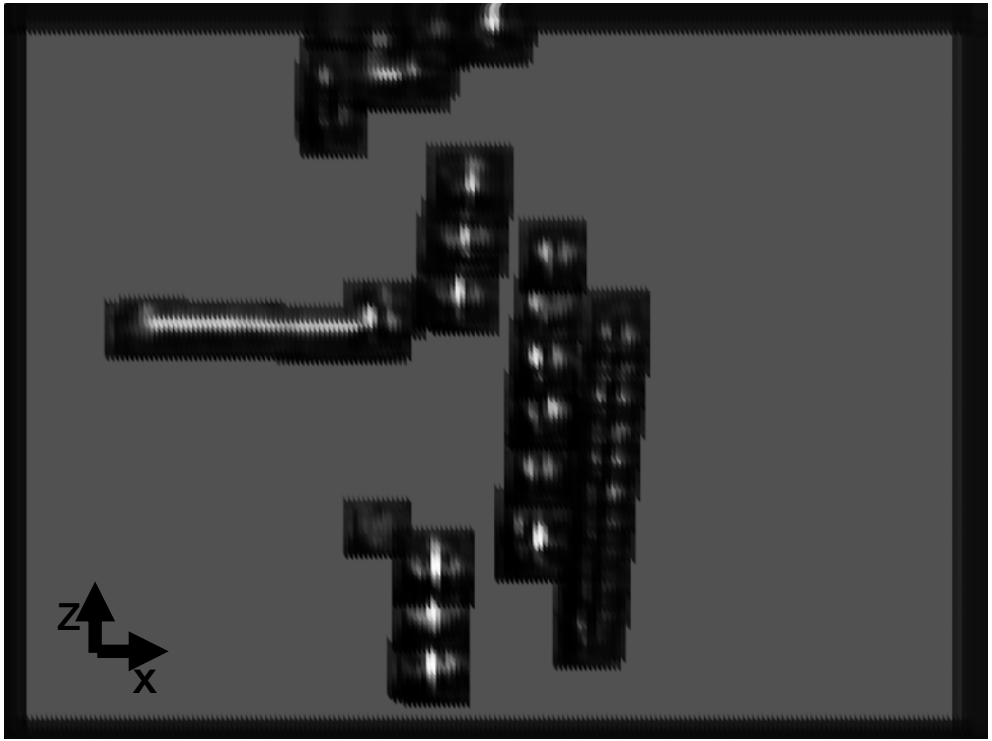
$$\theta_{\text{las}} \leq \omega_{\max} \cdot \Delta t$$

Optimized via dynamic programming

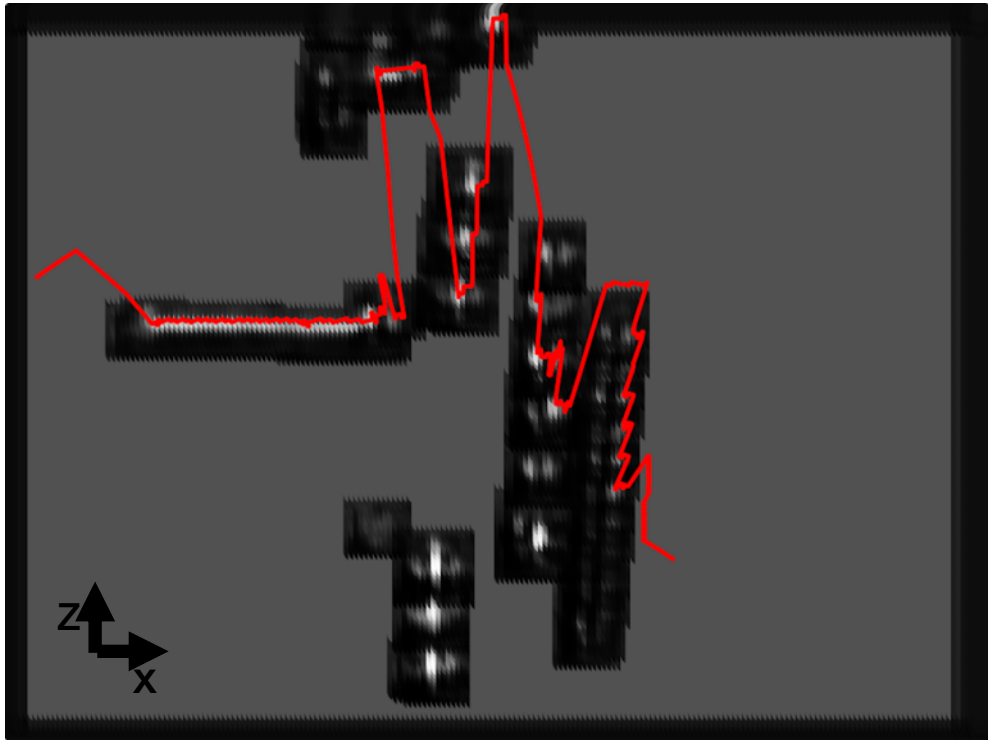
Constraint Graph



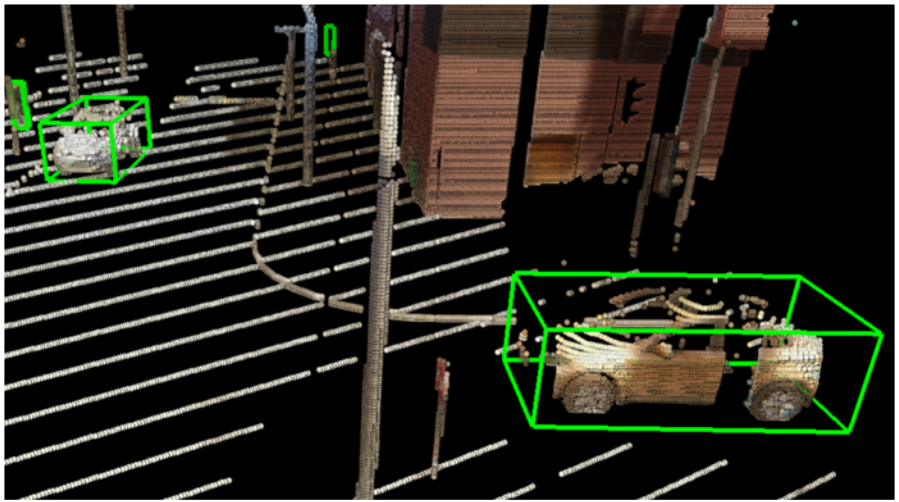
Uncertainty Map



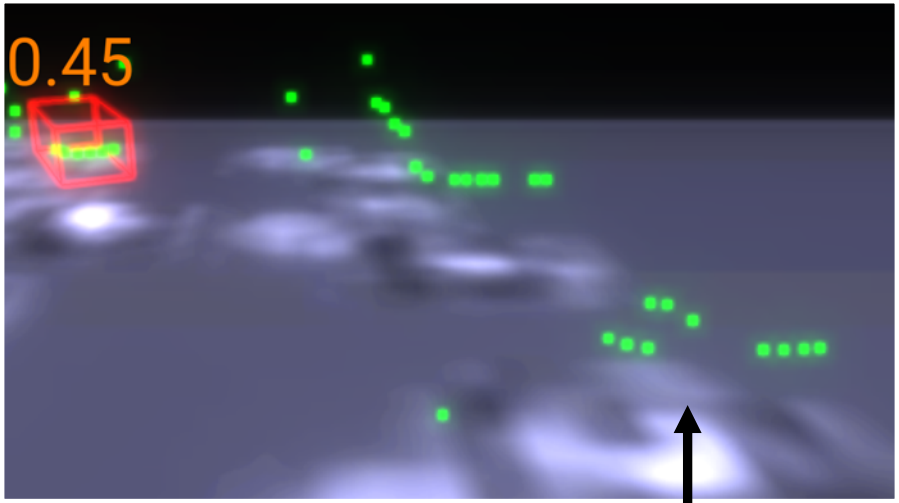
Optimized Light Curtain Placement

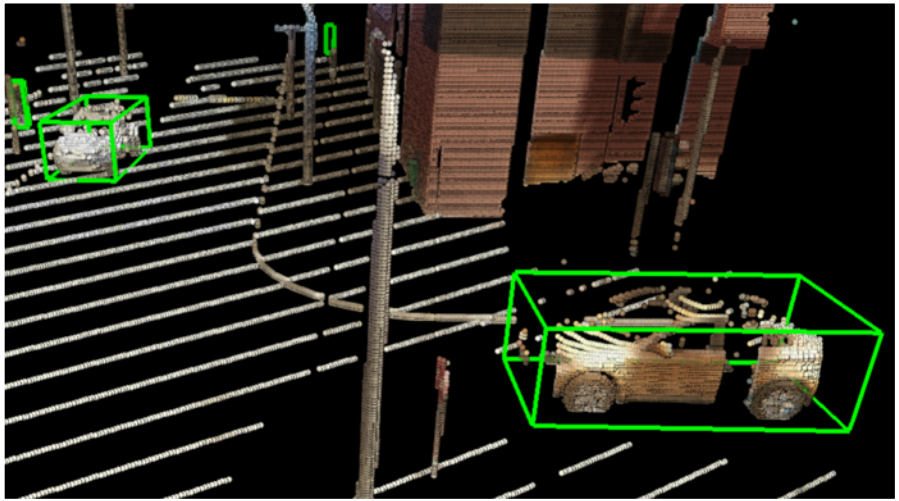


Detecting false negatives missed by LiDAR



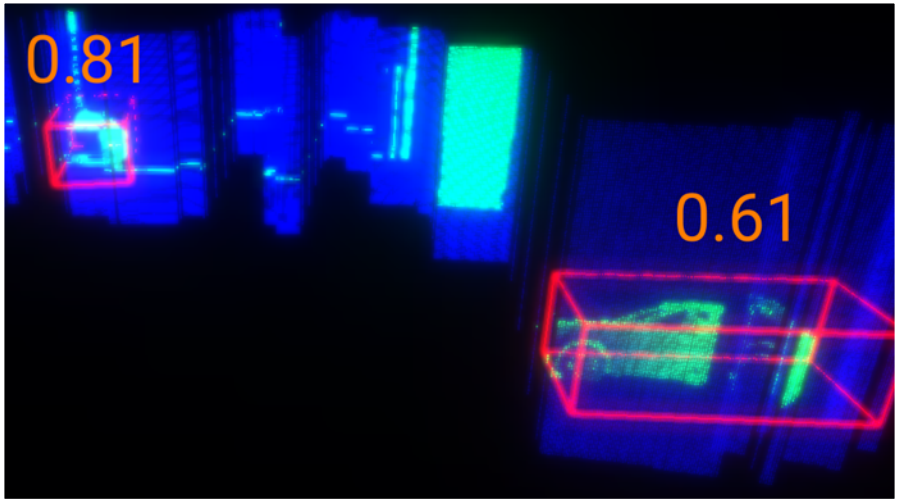
Detecting false negatives missed by LiDAR

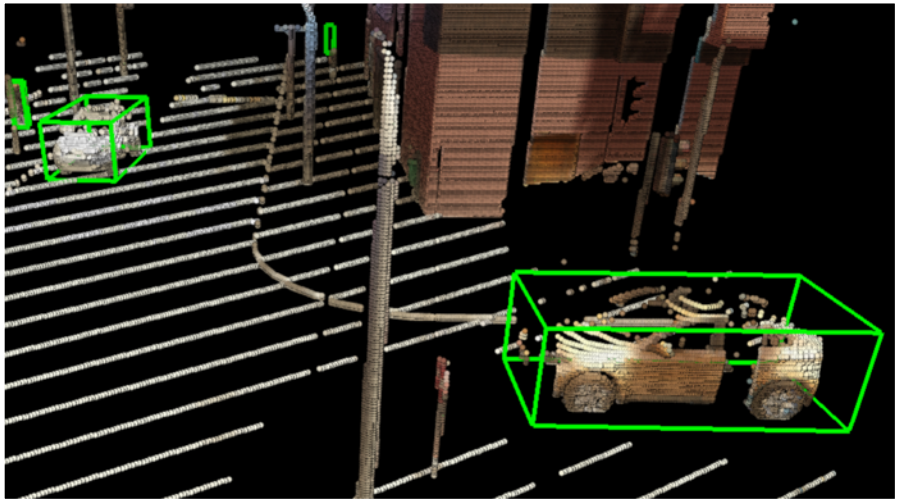




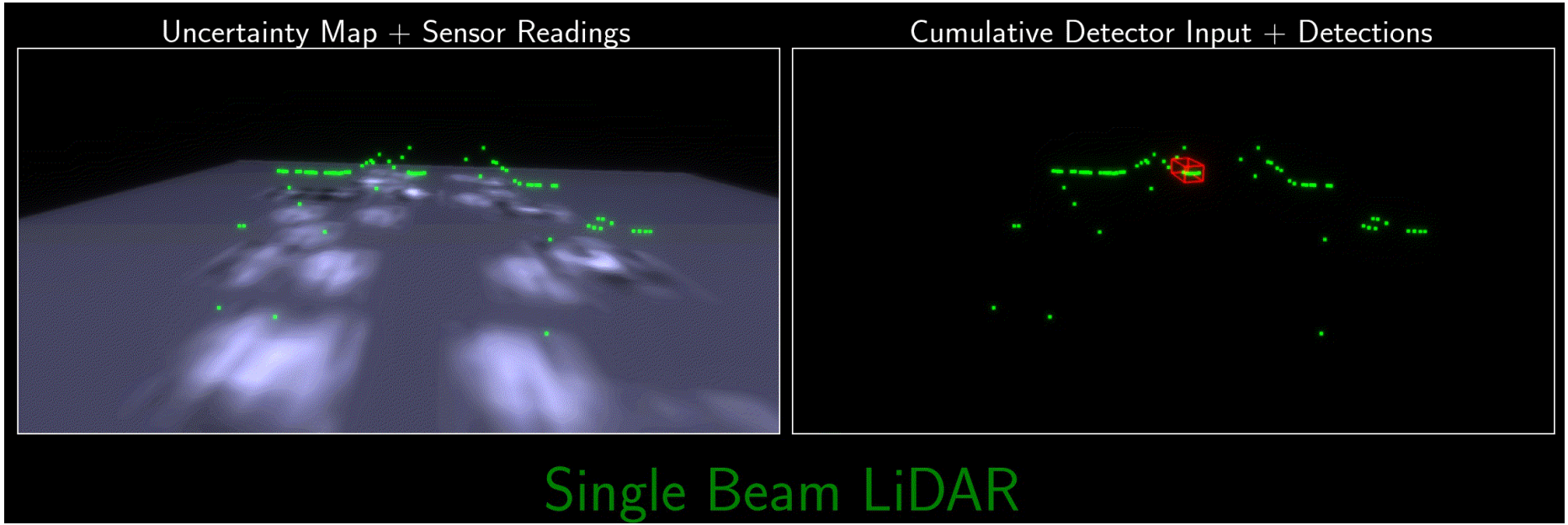
Missing detection

Detecting false negatives missed by LiDAR

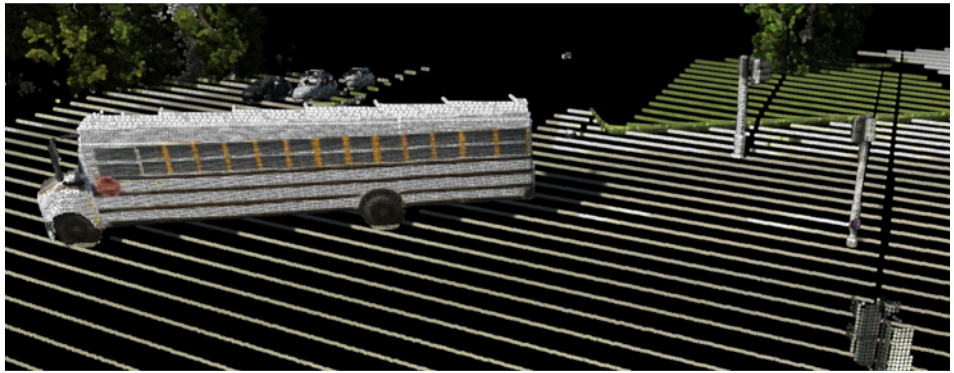




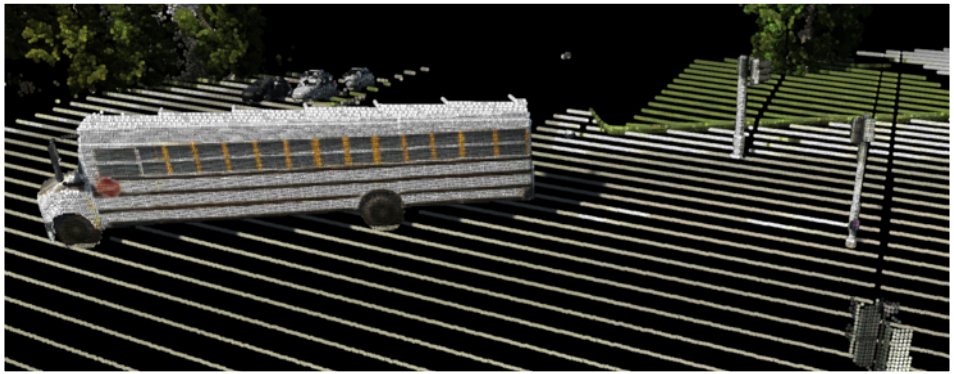
Detecting false negatives missed by LiDAR



Removing false positives detected by LiDAR

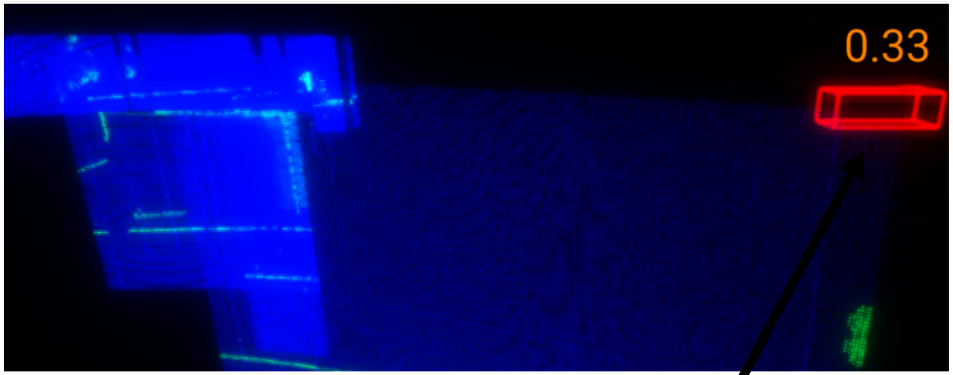


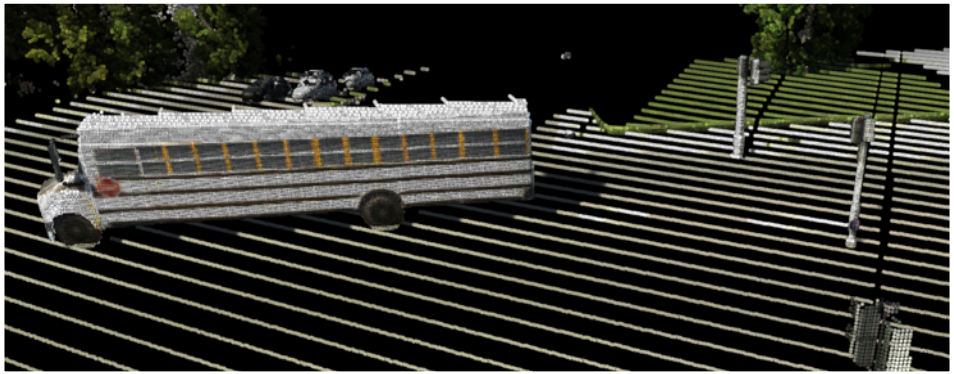
Removing false positives detected by LiDAR



False positive car detections

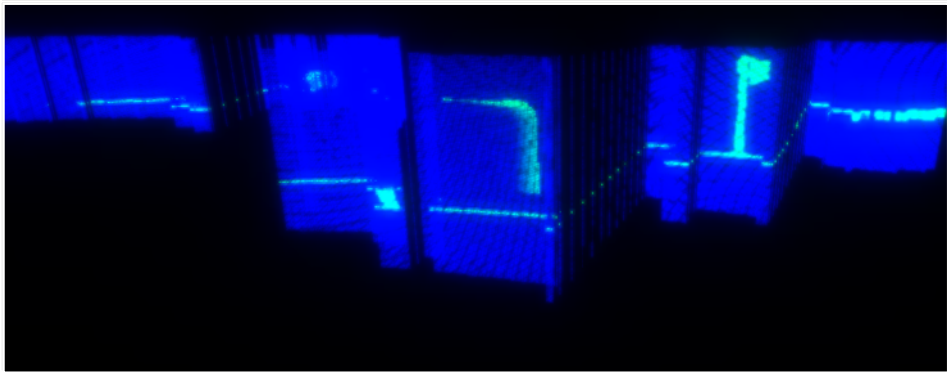
Removing false positives detected by LiDAR

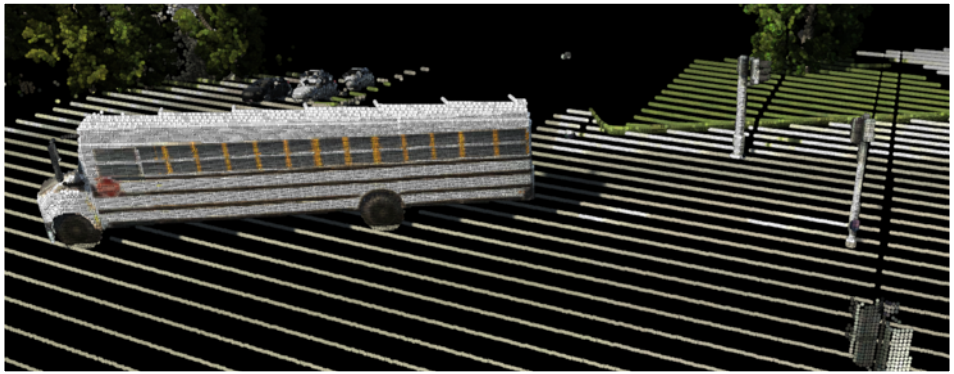




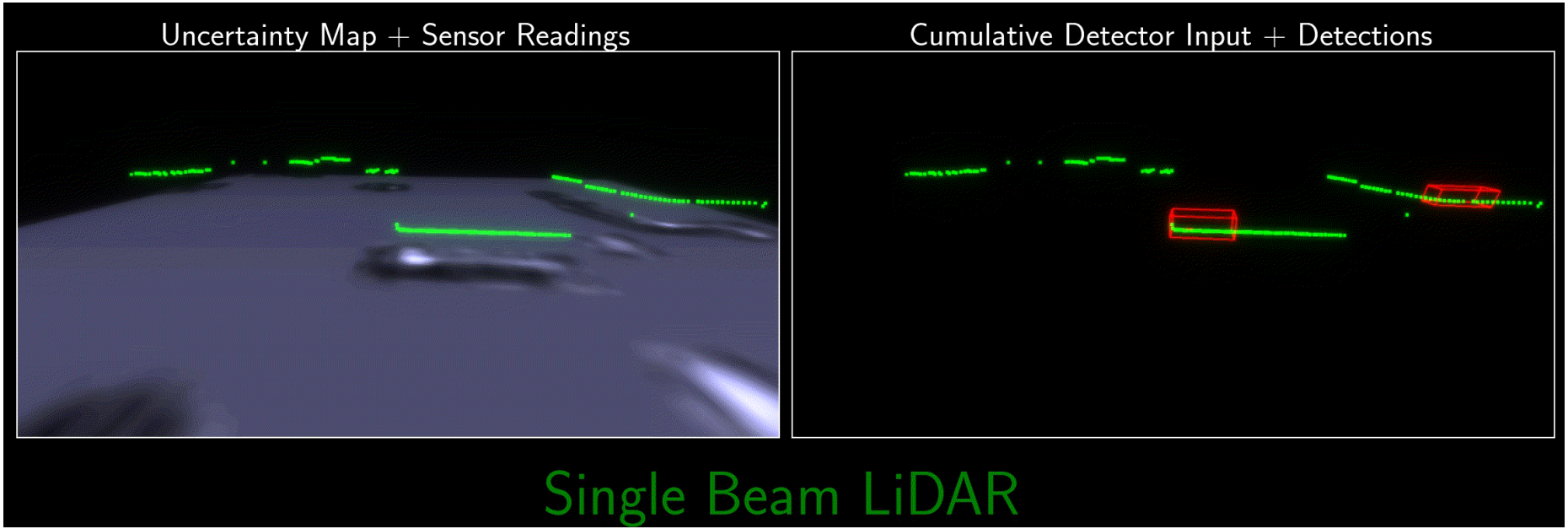
False positive car detection

Removing false positives detected by LiDAR

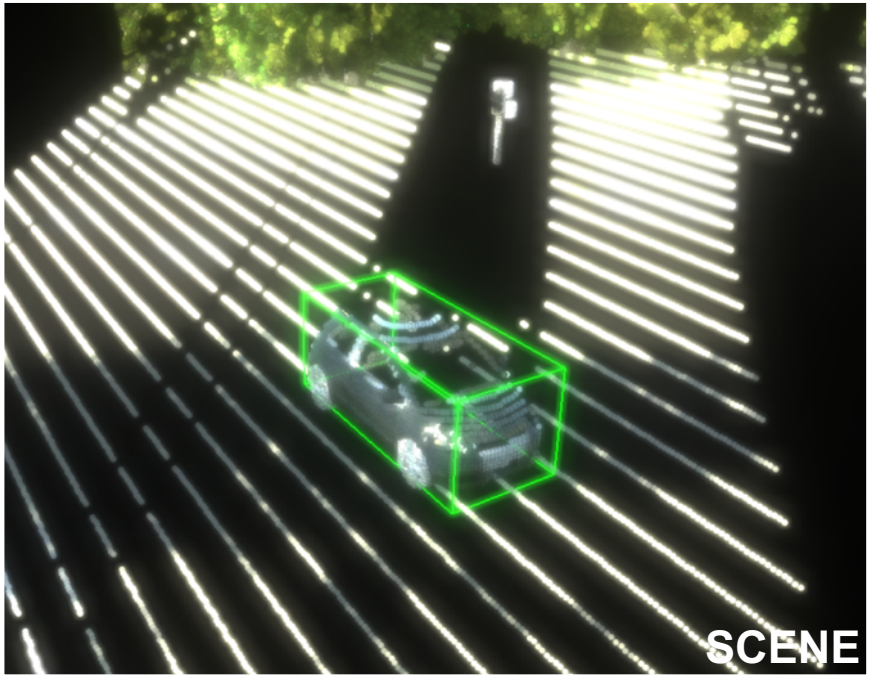




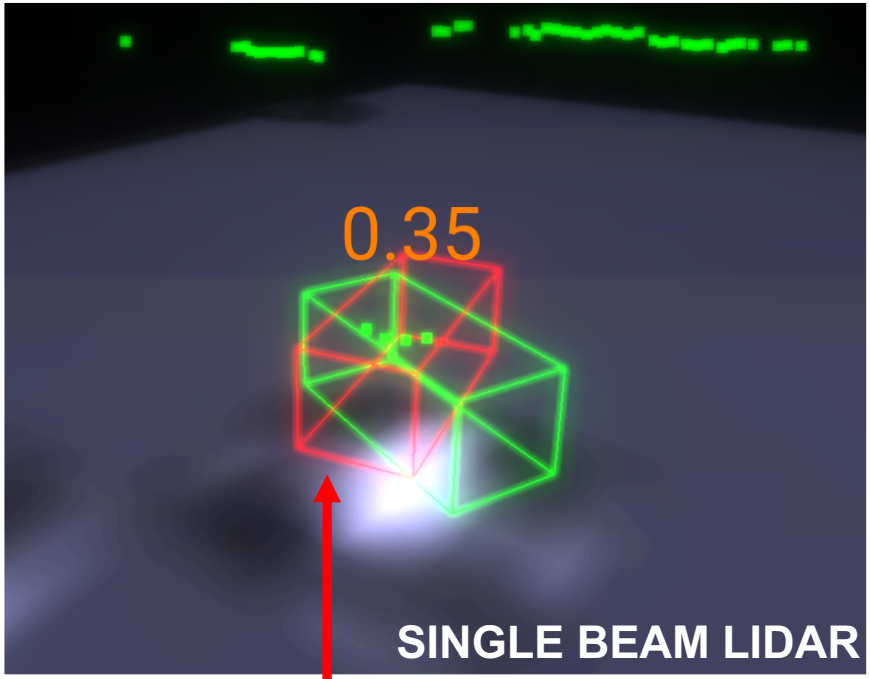
Removing false positives detected by LiDAR

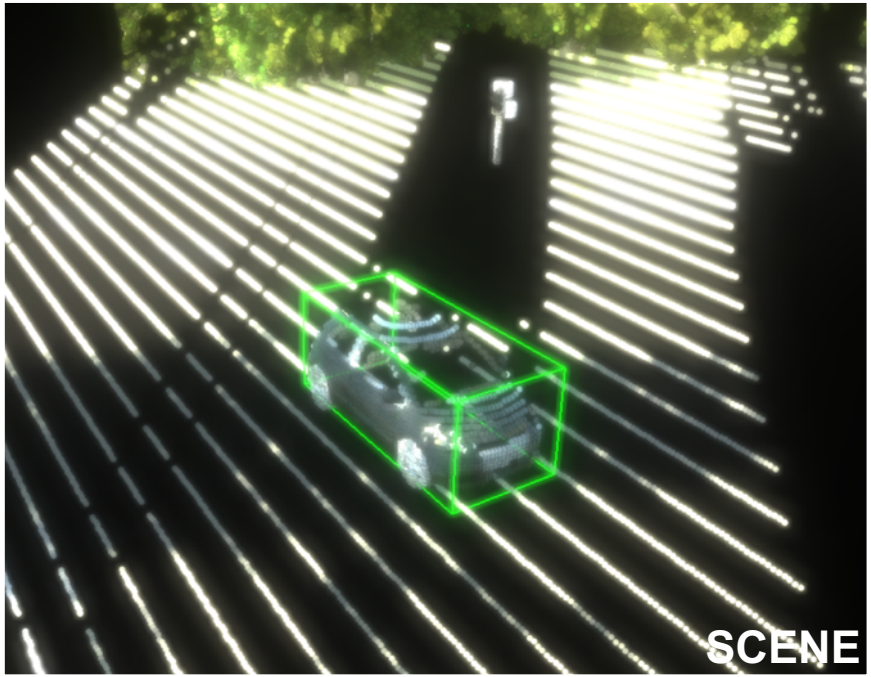


Correcting misaligned detection



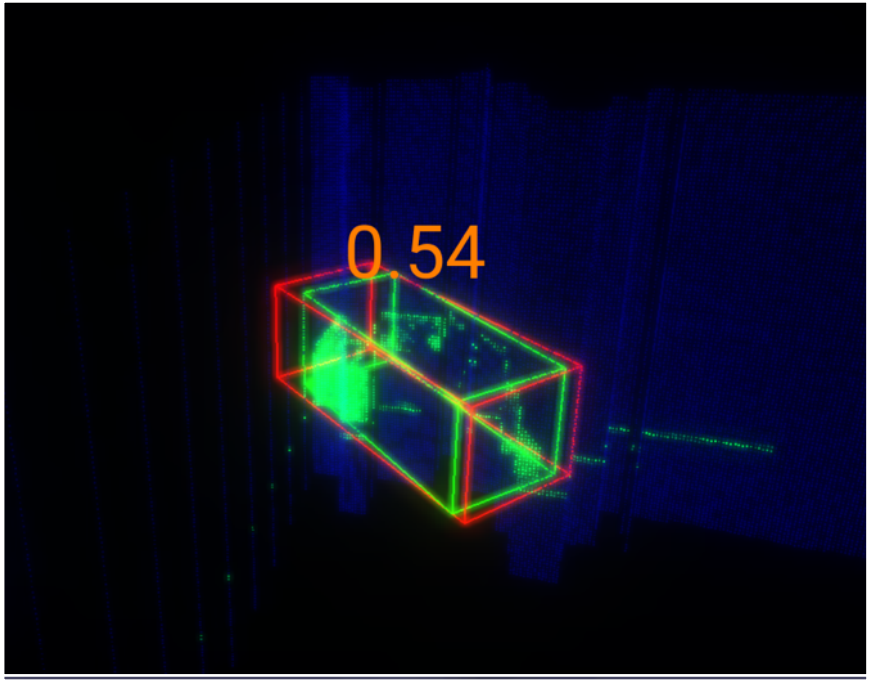
Correcting misaligned detection

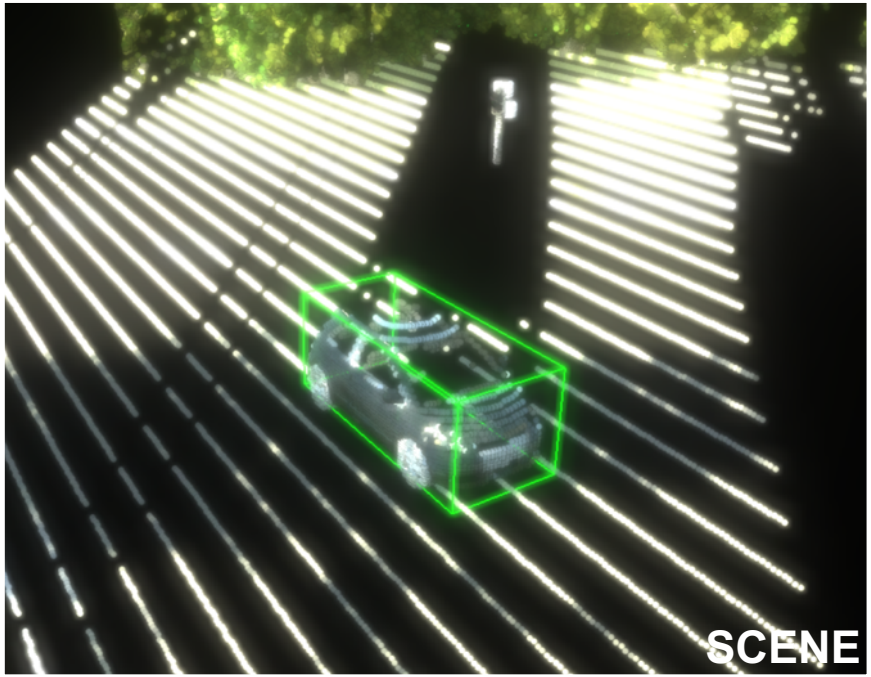




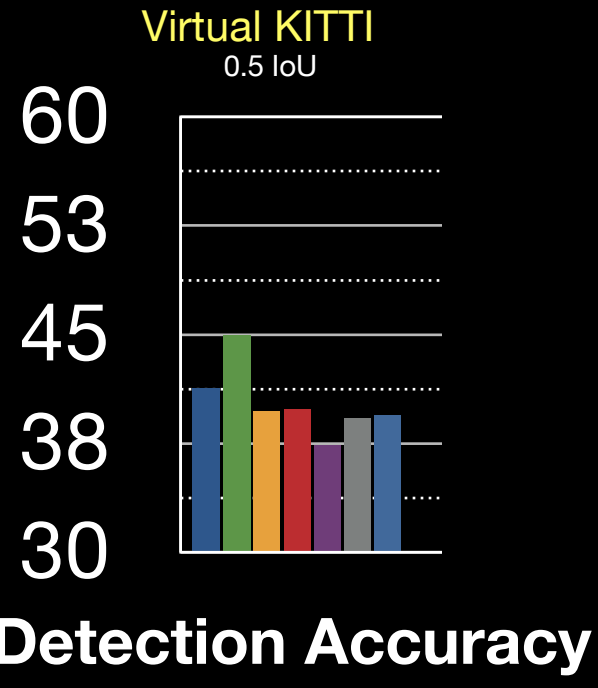
Misaligned detection

Correcting misaligned detection

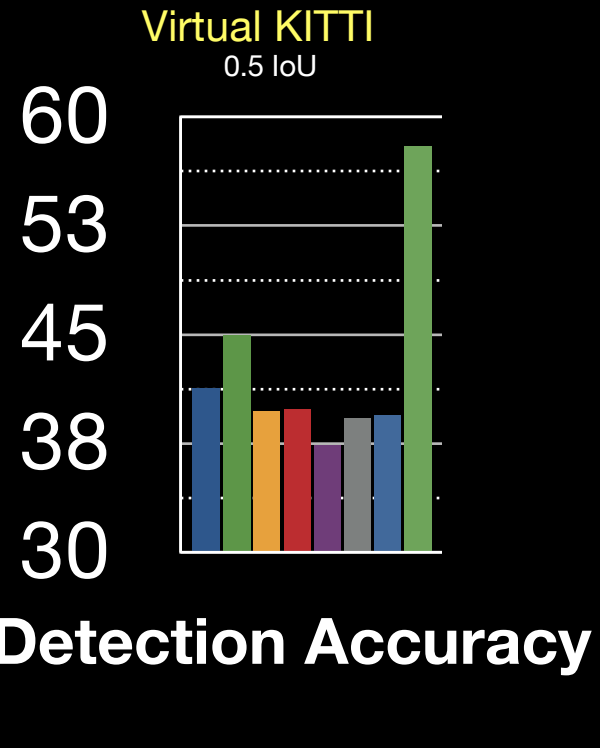




Comparison to baselines

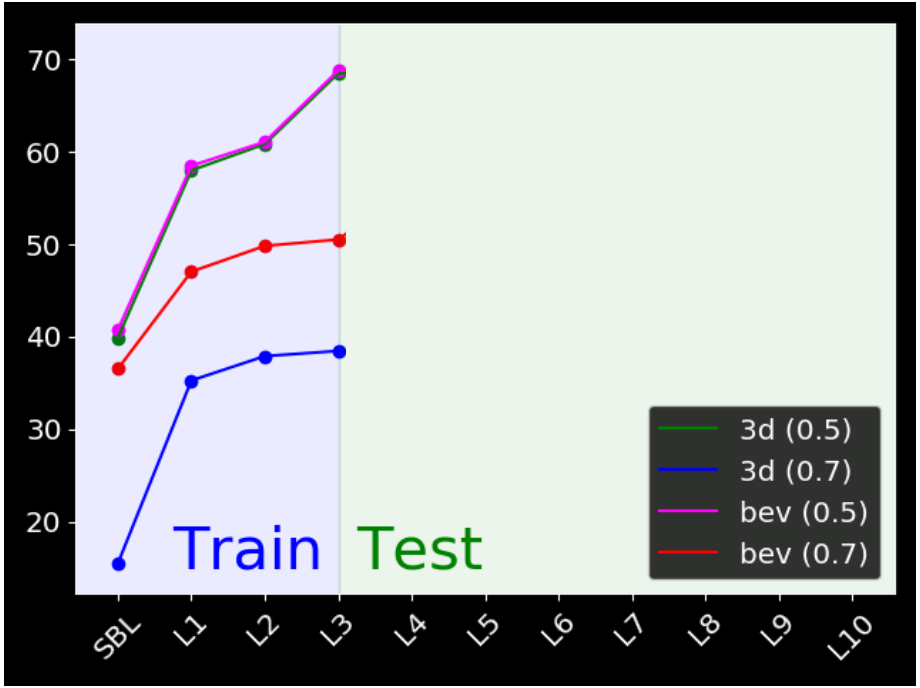


Comparison to baselines

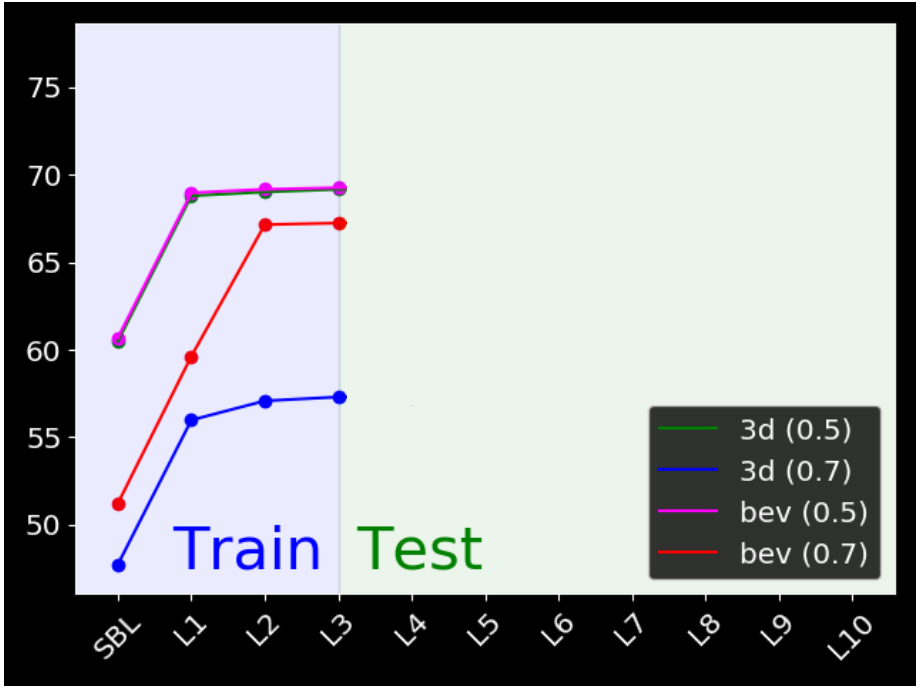


Performance generalizes to additional curtains

Generalization in Virtual KITTI

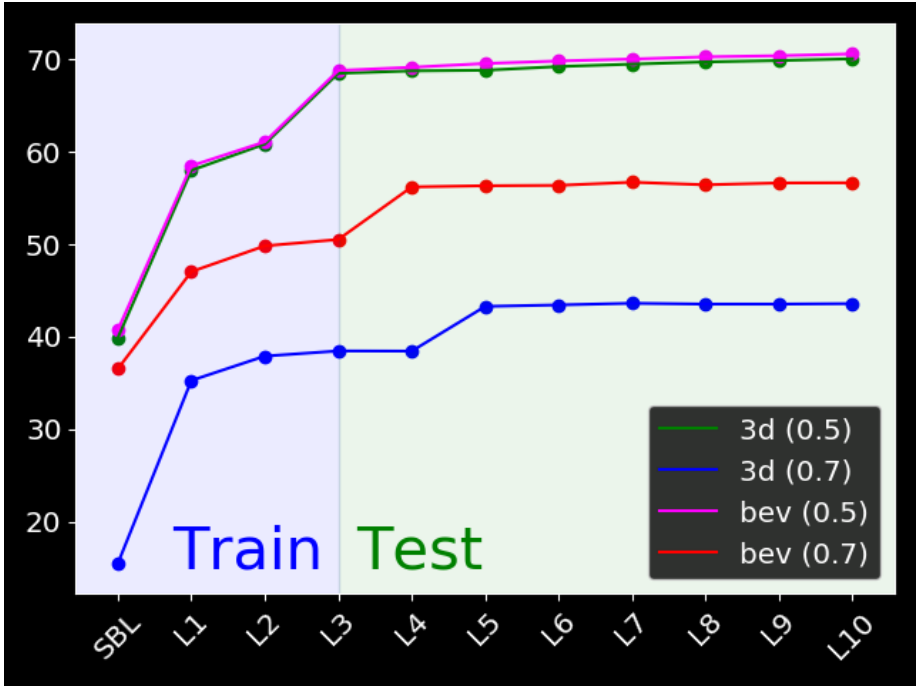


Generalization in SYNTHIA

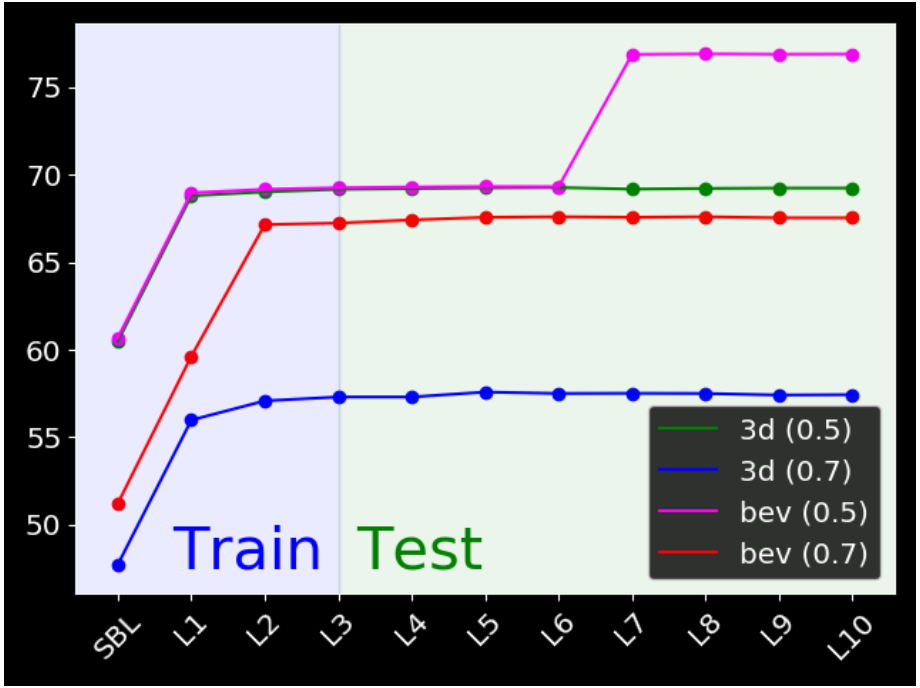


Performance generalizes to additional curtains

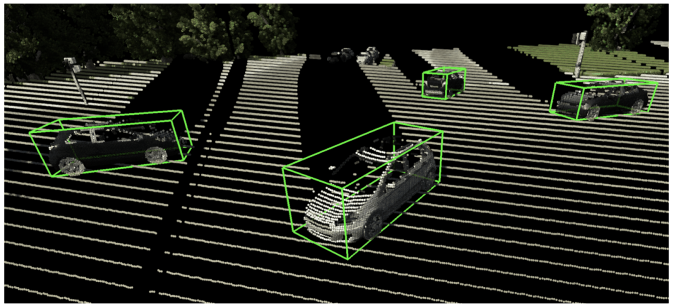
Generalization in Virtual KITTI

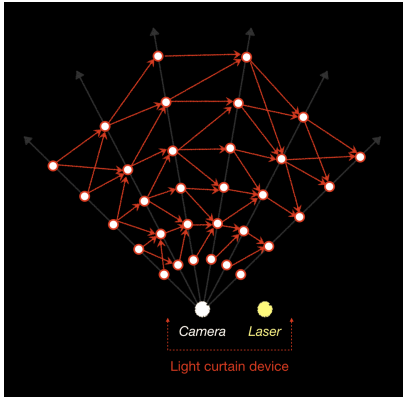


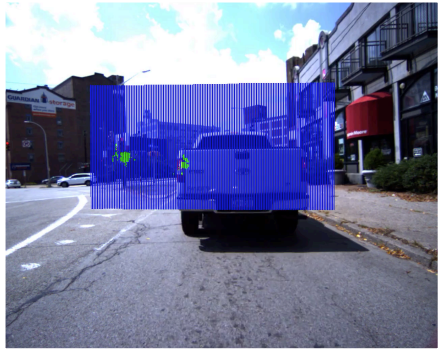
Generalization in SYNTHIA



Conclusions





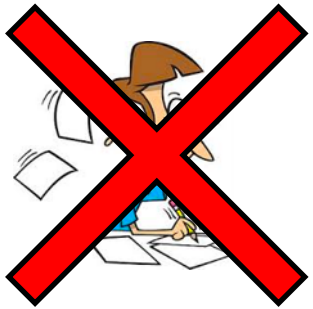


- We propose for a method for active detection using light curtains for autonomous driving.
- We optimize the light curtain placement by encoding the light curtain constrains into a constraint graph and using dynamic programming to maximize the objective.
- We show that our method can successively improve detection accuracy and is a step towards replacing expensive multi-beam LiDAR systems with inexpensive controllable sensors.

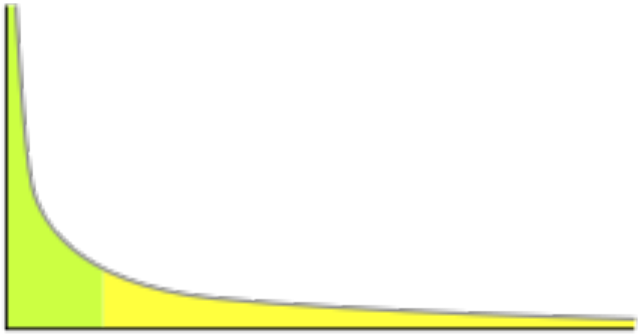
Self-Supervised Learning for Autonomous Driving

- **Active Perception with Light Curtains (ECCV 2020)**
- Self-supervised 3D Scene Flow (CVPR 2020)
- Self-supervised 3D Data Association (IROS 2020)

Sid Ancha



Human labeling



Regular, easy driving

Difficult driving

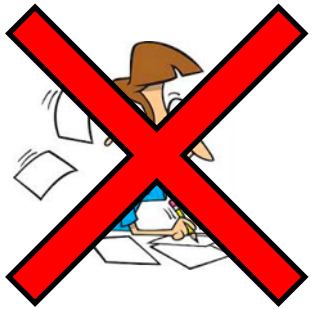
Weird driving

Self-Supervised Learning for Autonomous Driving

- Active Perception with Light Curtains (ECCV 2020)
- **Self-supervised 3D Scene Flow (CVPR 2020)**
- Self-supervised 3D Data Association (IROS 2020)

Himangi Mittal

Brian Okorn



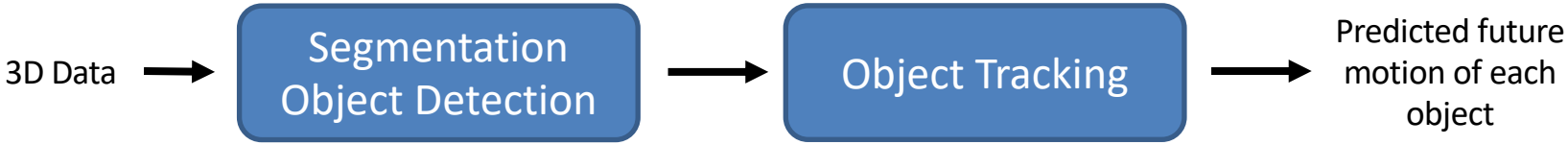
Human labeling

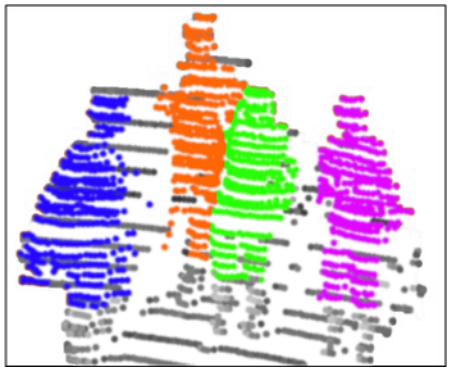
Regular, easy driving

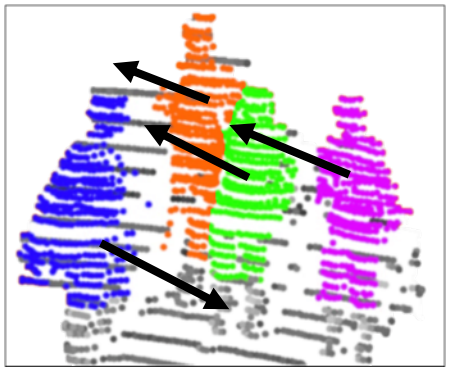
Difficult driving

Weird driving

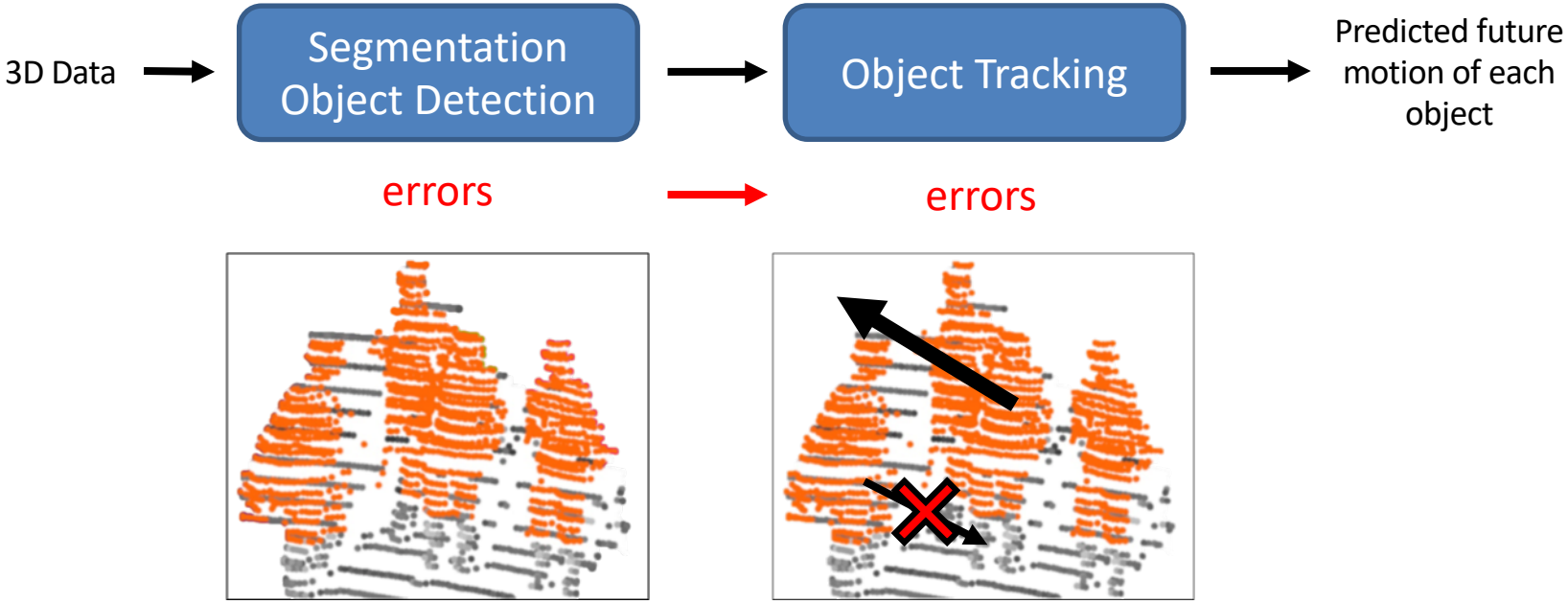
Traditional Tracking Pipeline



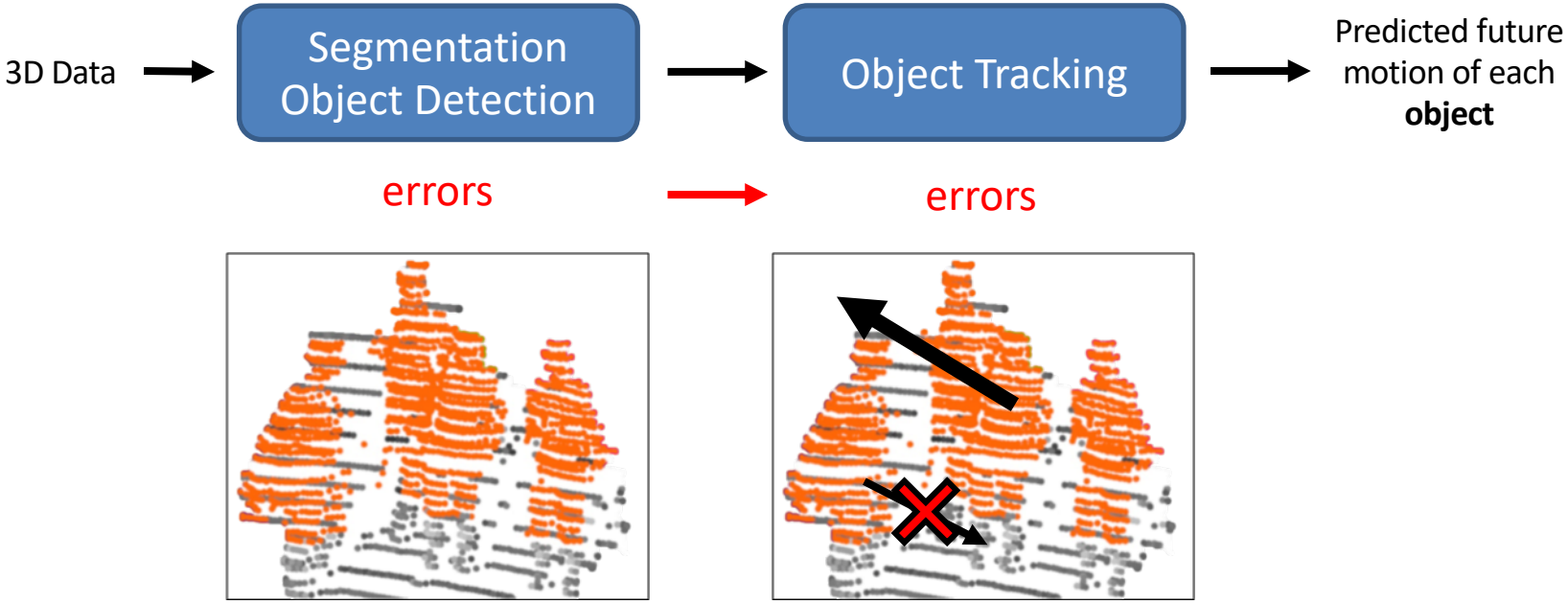




Traditional Tracking Pipeline

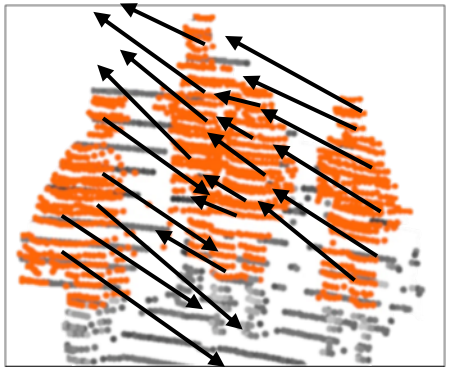


Traditional Tracking Pipeline

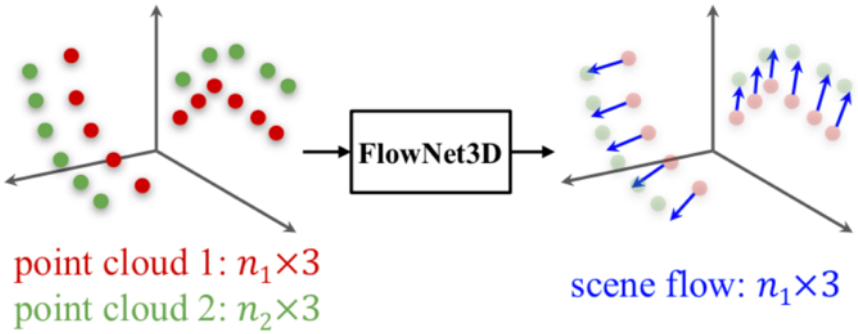


Scene Flow Pipeline

3D Data → 3D Scene Flow → Predicted future motion of each **point**



Previous Work: Deep Learning for Flow

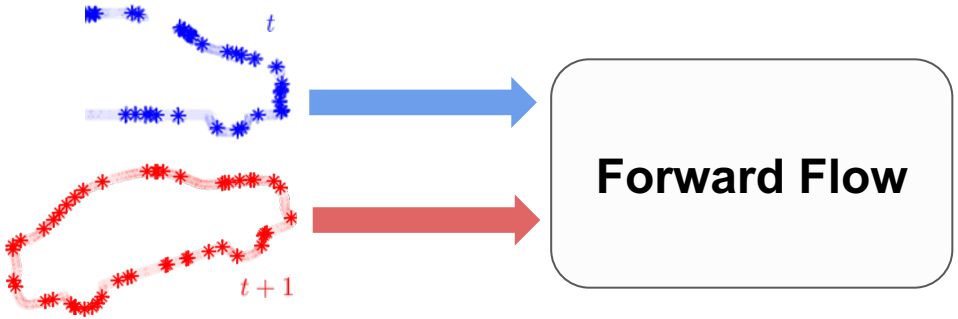


+ Accurate Deep Learned Scene Flow Estimation

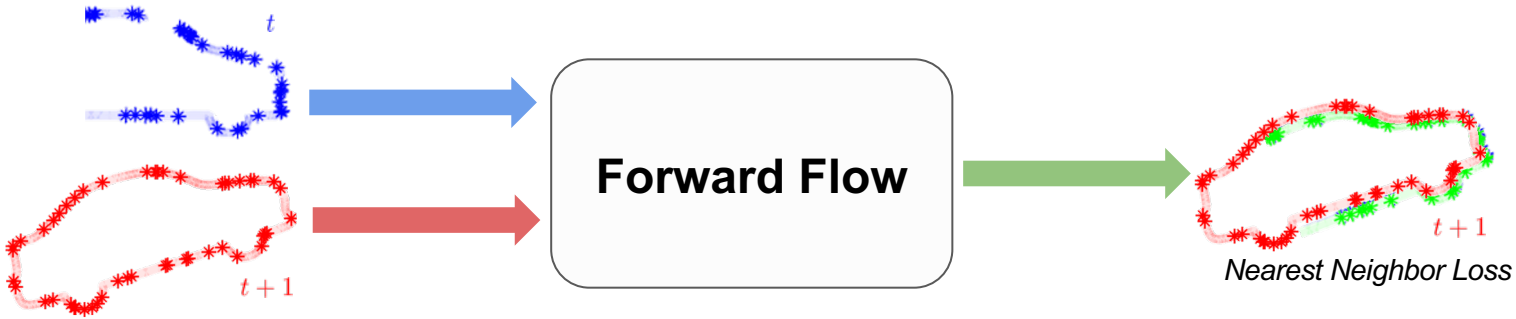
- Requires Point-Level Motion Labels to Train

Liu, Xingyu, Charles R. Qi, and Leonidas J. Guibas. "Learning Scene Flow in 3D Point Clouds." *CVPR* 2019.

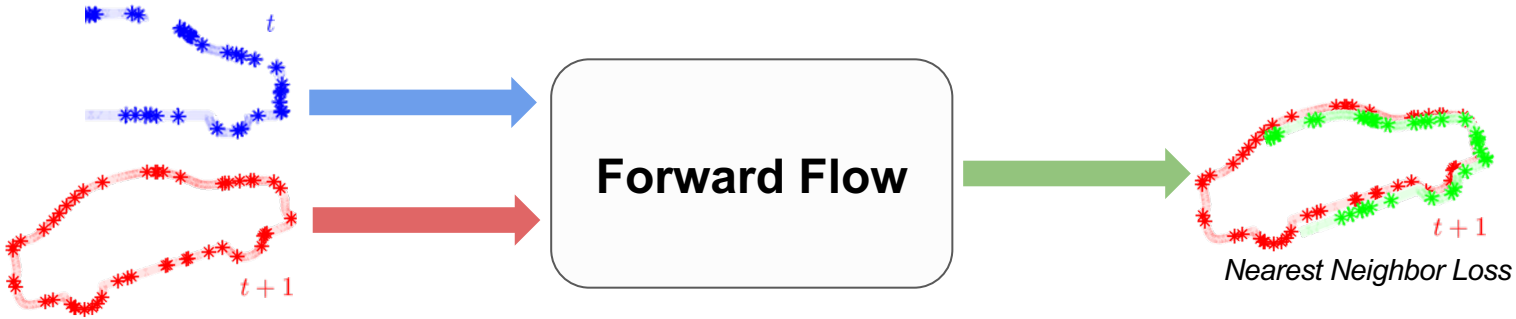
Our Approach: Self-supervised Scene Flow



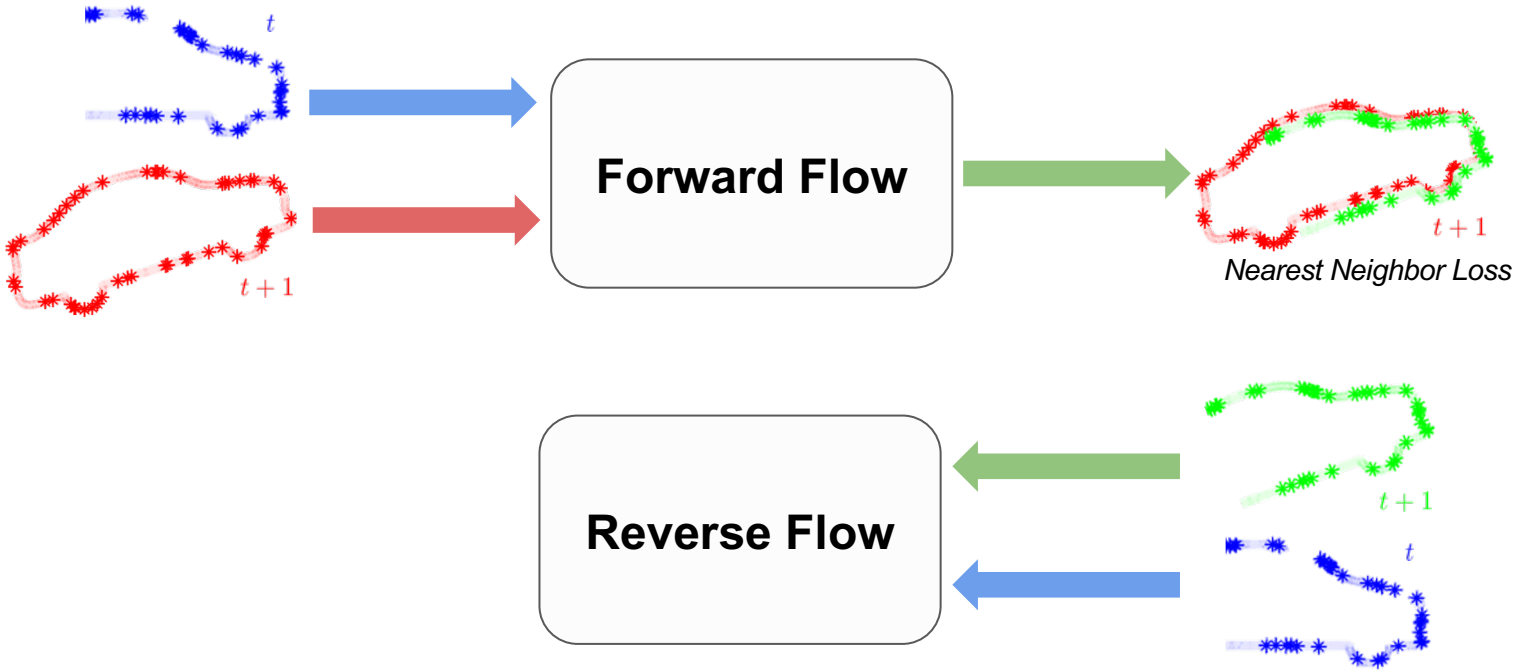
Our Approach: Self-supervised Scene Flow



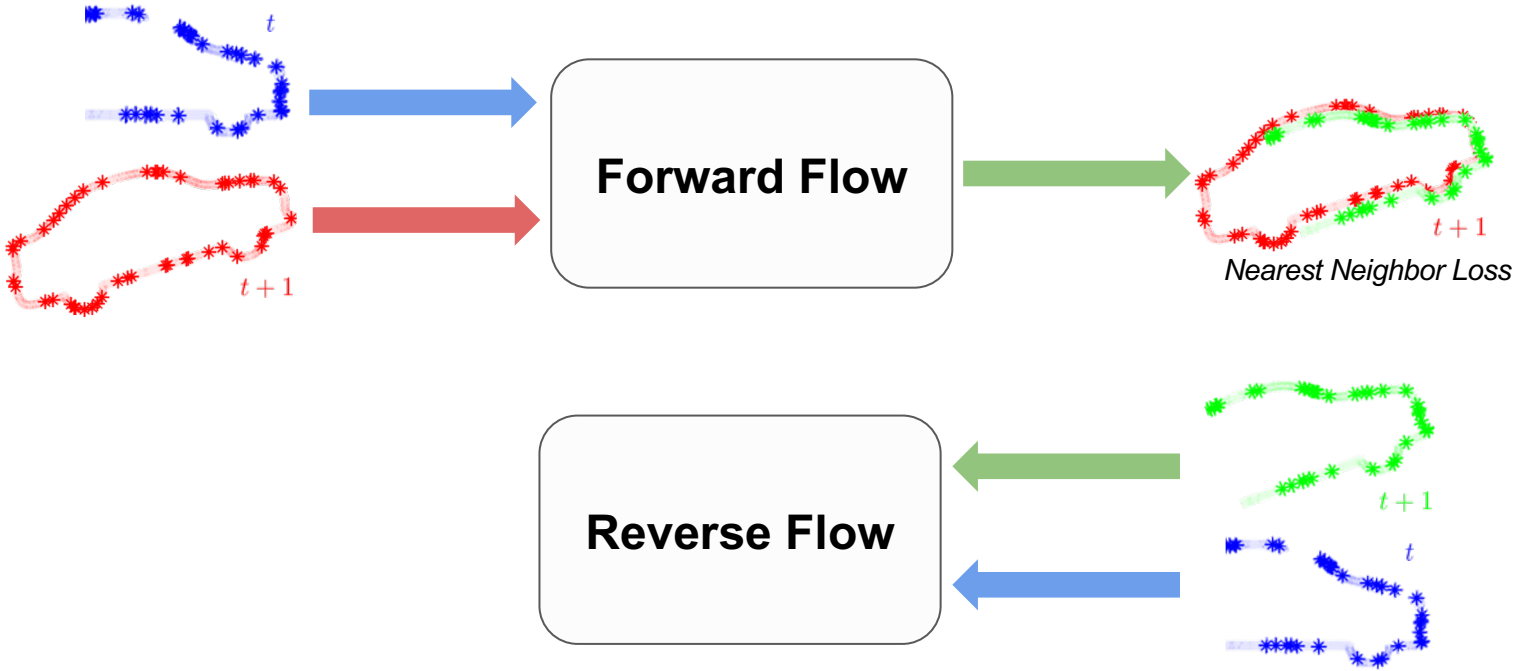
Our Approach: Self-supervised Scene Flow



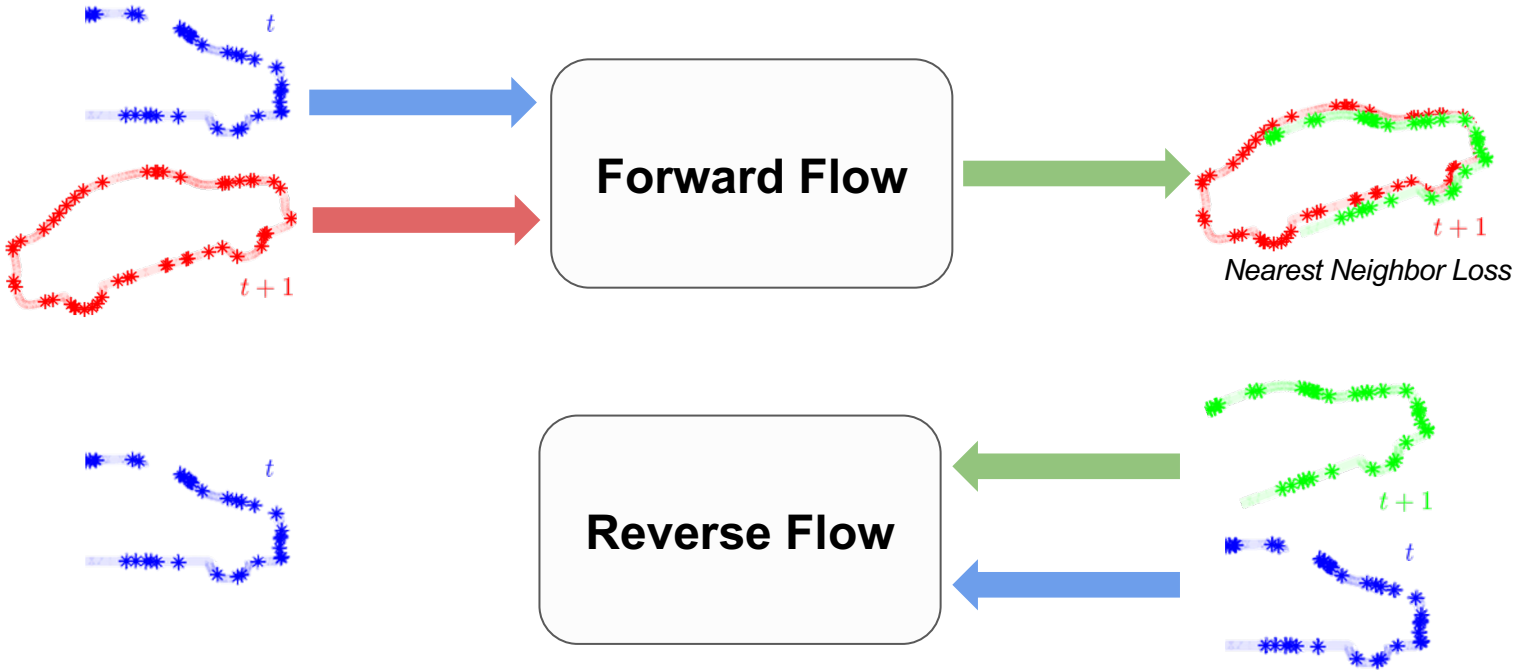
Our Approach: Self-supervised Scene Flow



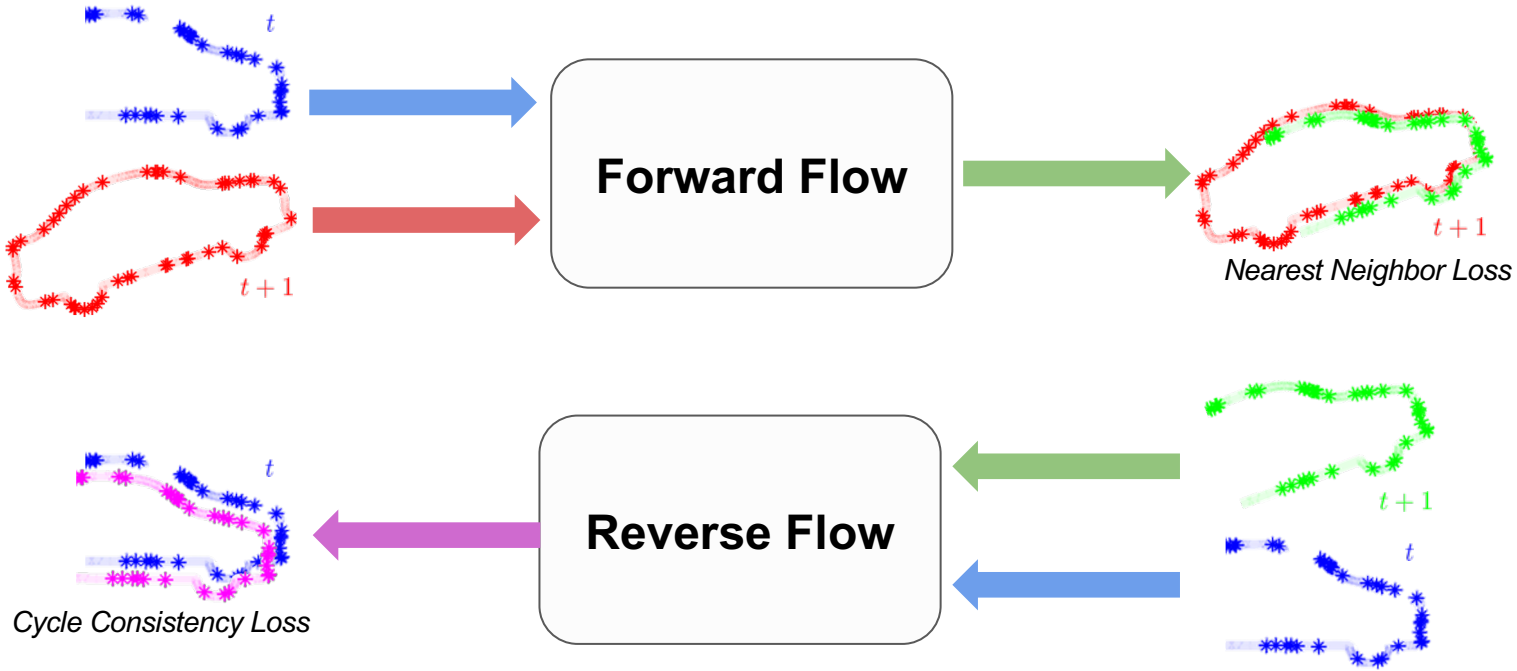
Our Approach: Self-supervised Scene Flow



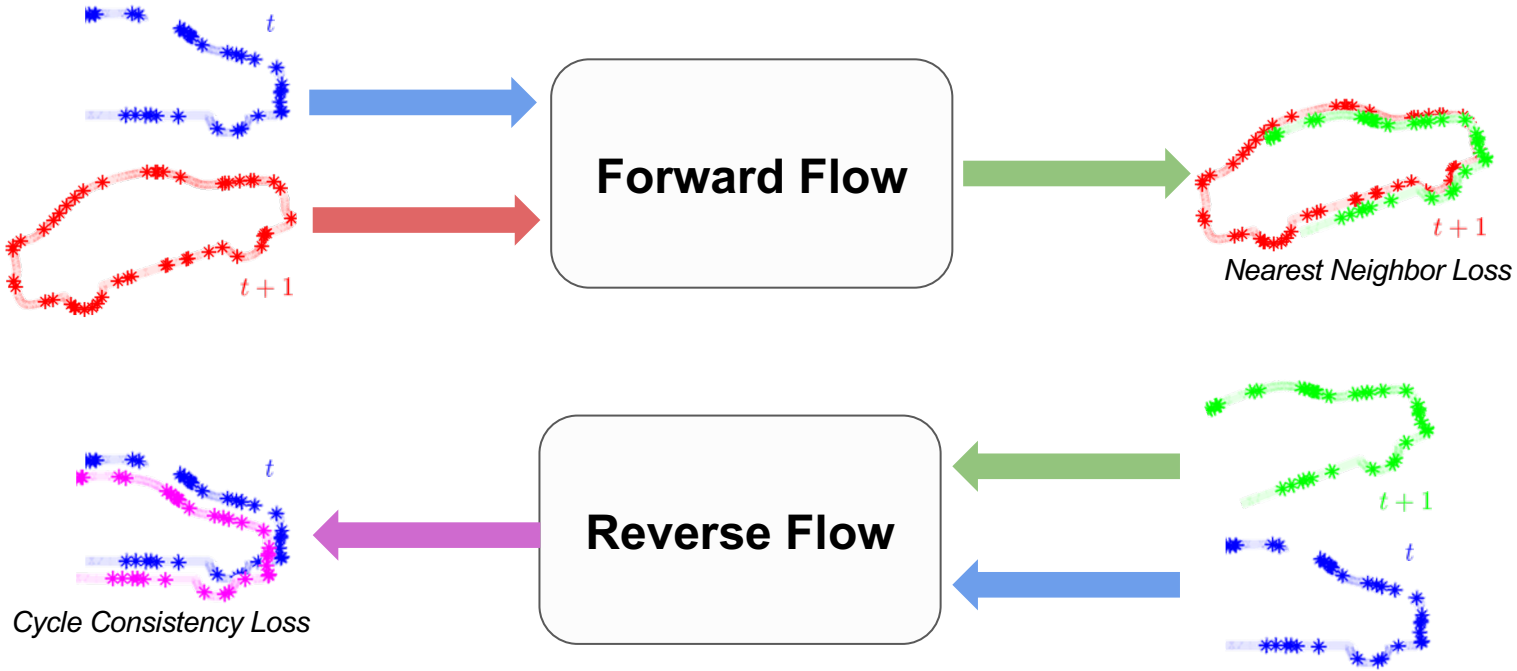
Our Approach: Self-supervised Scene Flow



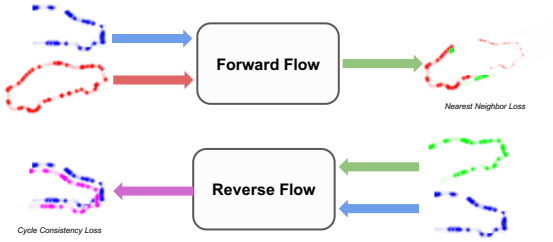
Our Approach: Self-supervised Scene Flow



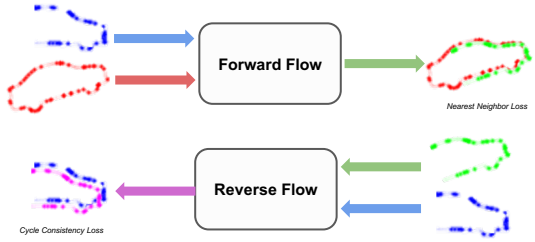
Our Approach: Self-supervised Scene Flow

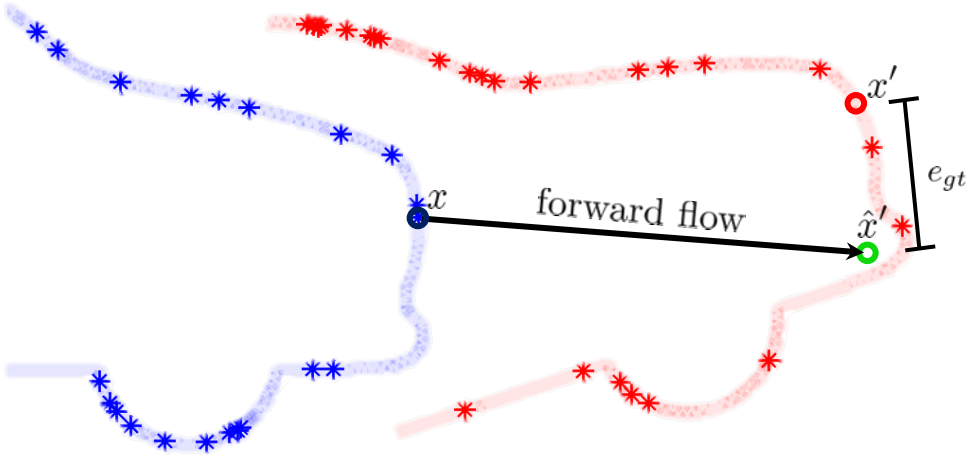


Nearest Neighbor Loss



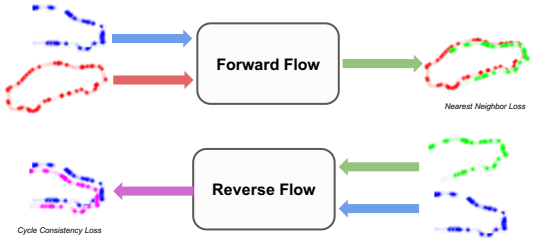
Nearest Neighbor Loss

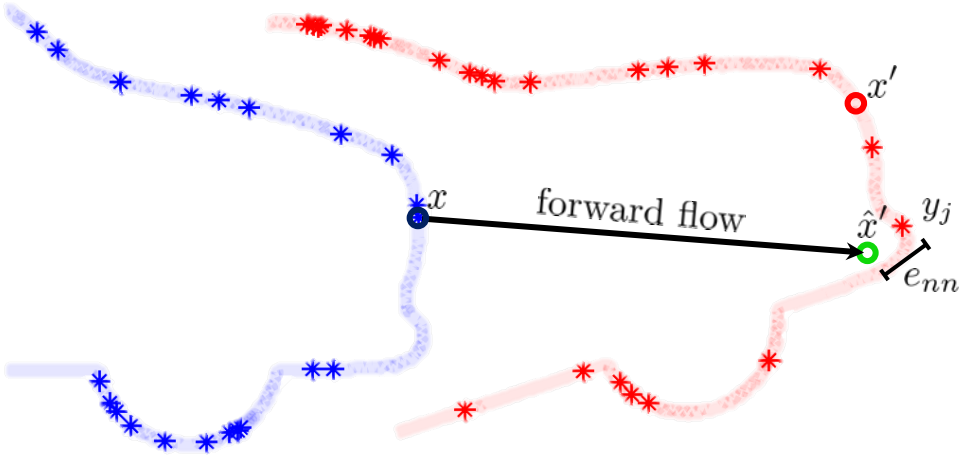




$$\mathcal{L}_{gt} = \frac{1}{N} \sum_{i}^{N} \|\hat{x}'_i - x'_i\|^2$$

Nearest Neighbor Loss

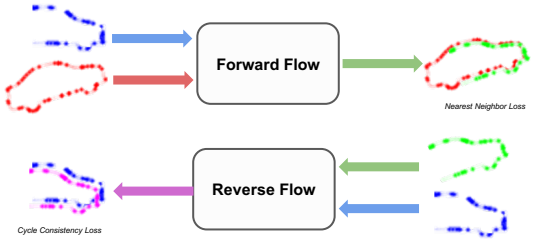


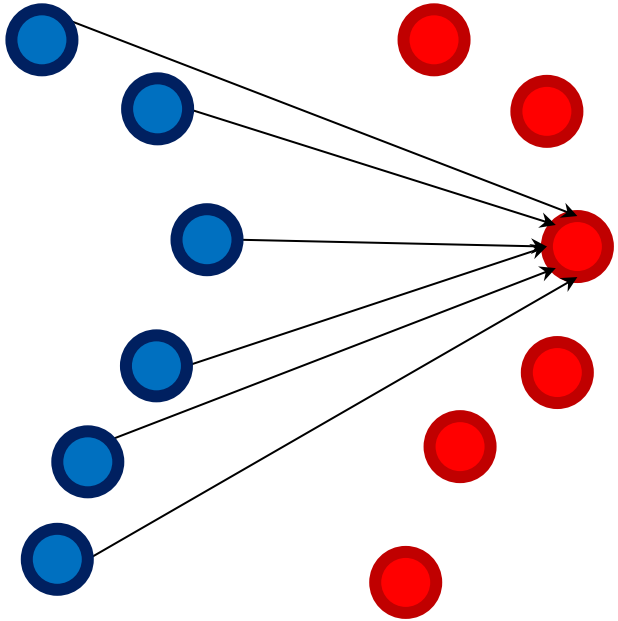


$$\mathcal{L}_{nn} = \frac{1}{N} \sum_{i}^{N} \min_{y_j \in \mathcal{Y}} \|\hat{x}'_i - y_j\|^2$$

Nearest Neighbor Loss

Degenerate Solution



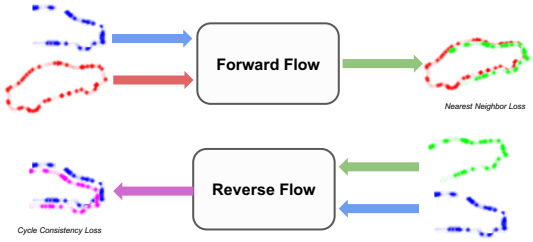


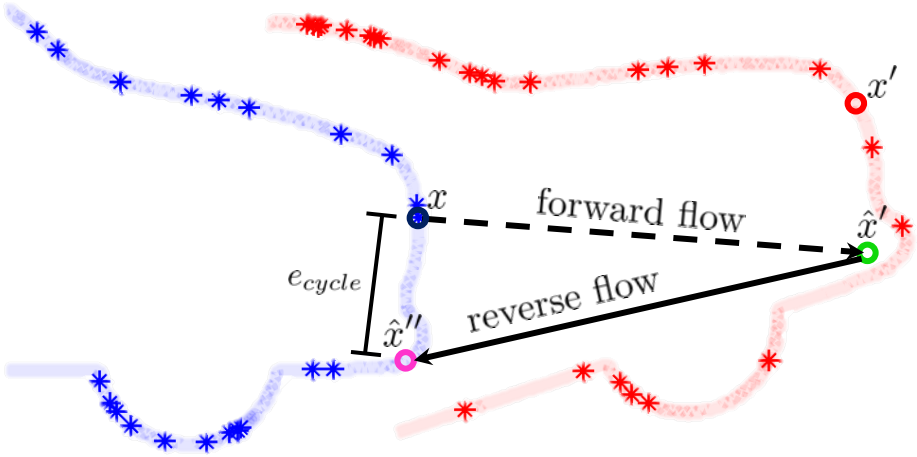
$$\mathcal{L}_{nn} = \frac{1}{N} \sum_{i}^{N} \min_{y_j \in \mathcal{Y}} \|\hat{x}'_i - y_j\|^2$$

Distance to nearest neighbor = 0

Nearest Neighbor Loss = 0

Cycle Consistency Loss

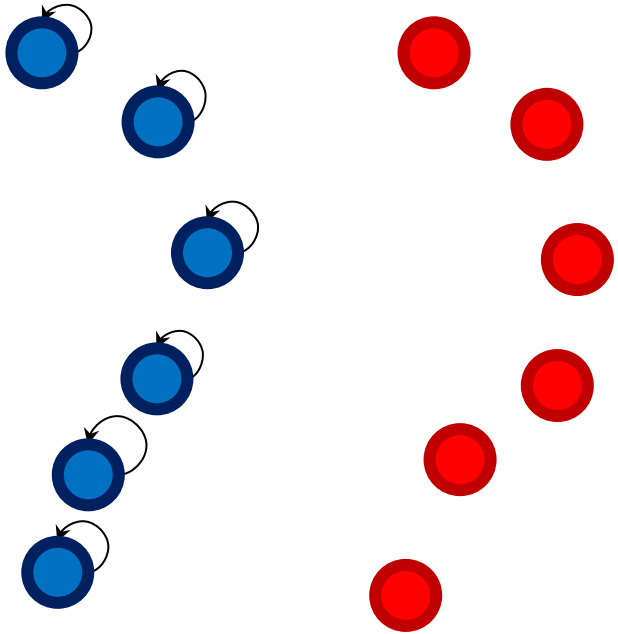


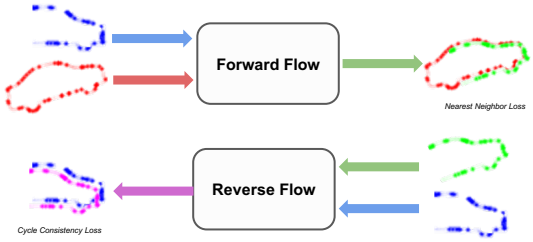


$$\mathcal{L}_{cycle} = \frac{1}{N}\sum_{i}^{N} \|\hat{x}_i'' - x_i\|^2$$

Cycle Consistency Loss

Degenerate Solution





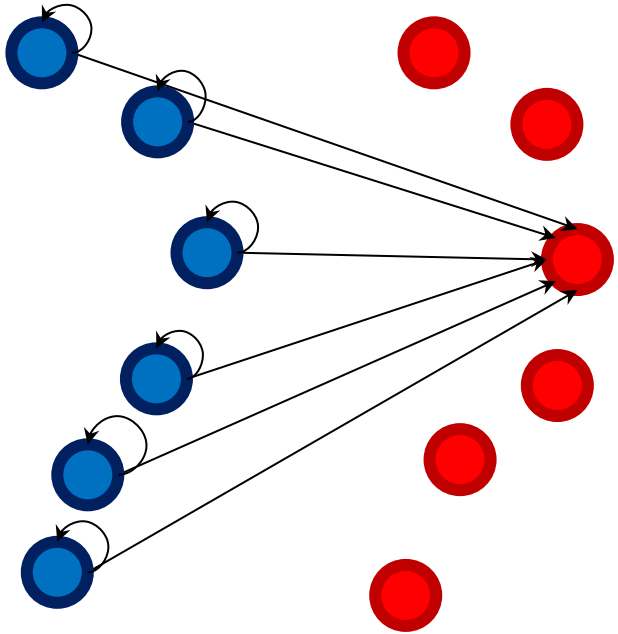
$$\mathcal{L}_{cycle} = \frac{1}{N}\sum_{i}^{N} \|\hat{x}_i'' - x_i\|^2$$

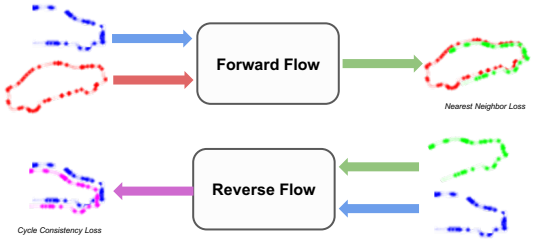
Zero forward or backward flow

Cycle Consistency Loss = 0

Self-Supervised Losses

Conflicting Degenerate Solution

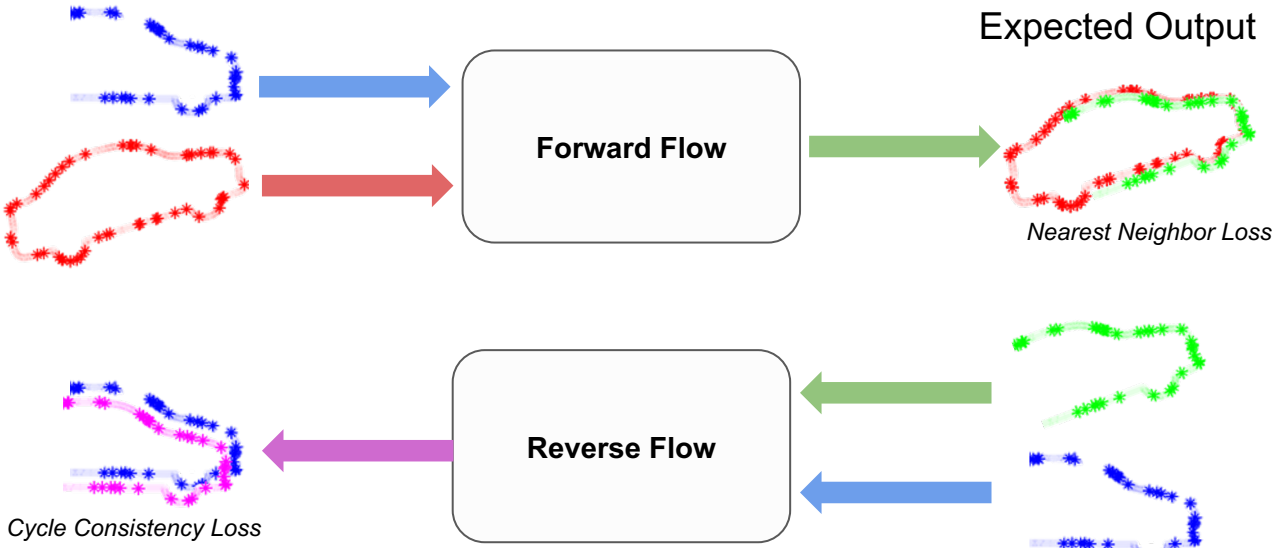




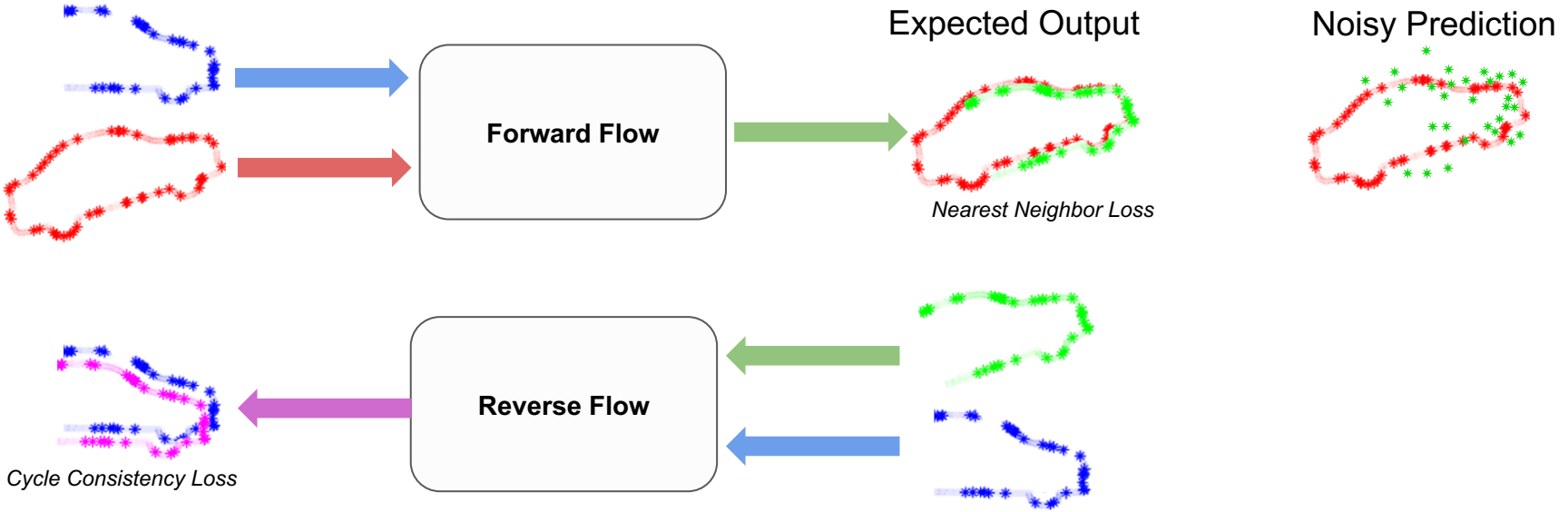
$$\mathcal{L}_{nn} = \frac{1}{N} \sum_{i}^{N} \min_{y_j \in \mathcal{Y}} \|\hat{x}'_i - y_j\|^2$$

$$\mathcal{L}_{cycle} = \frac{1}{N} \sum_{i}^{N} \|\hat{x}''_i - x_i\|^2$$

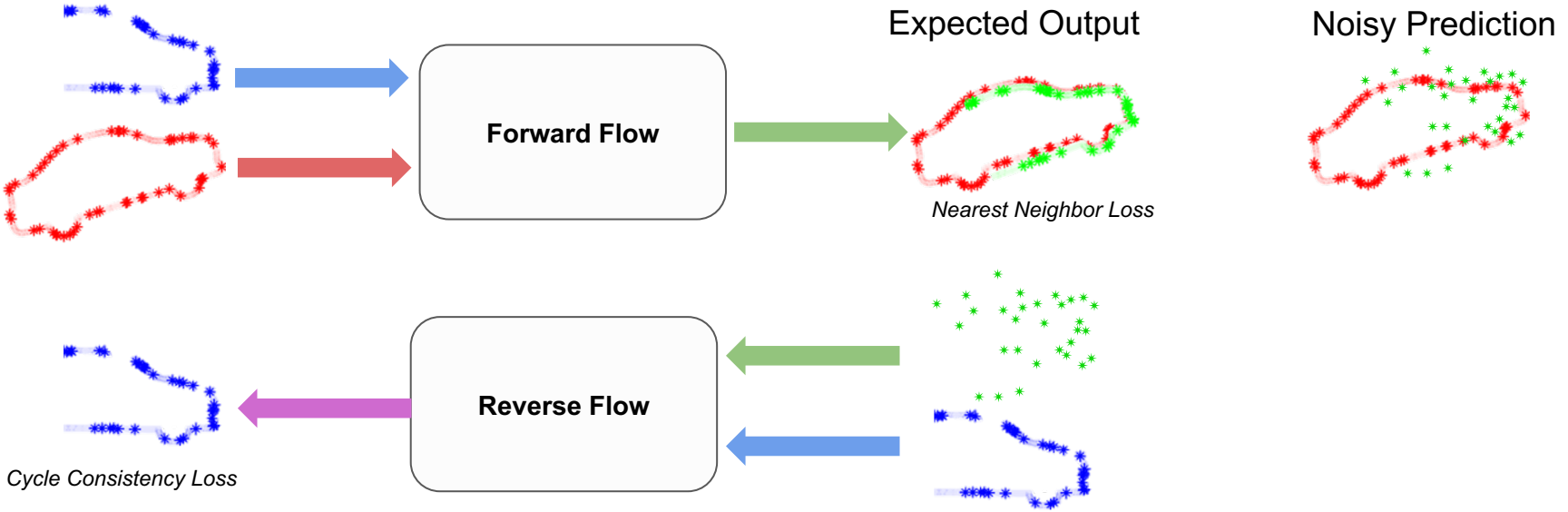
Self-Supervised Scene Flow



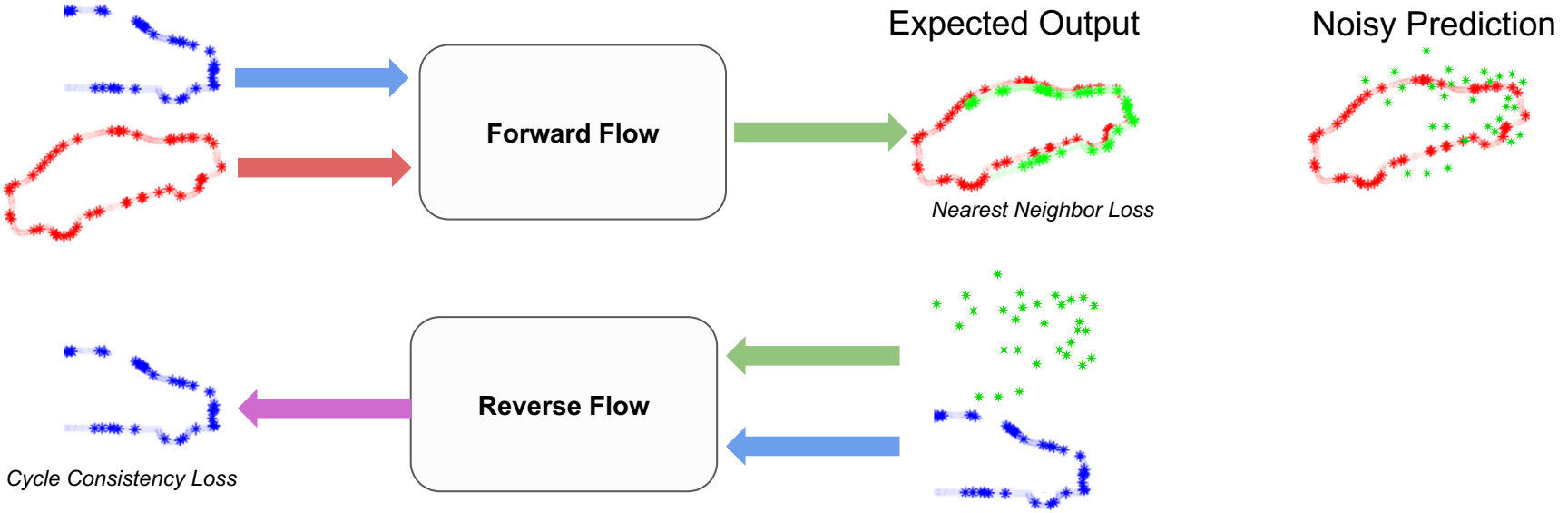
Self-Supervised Scene Flow



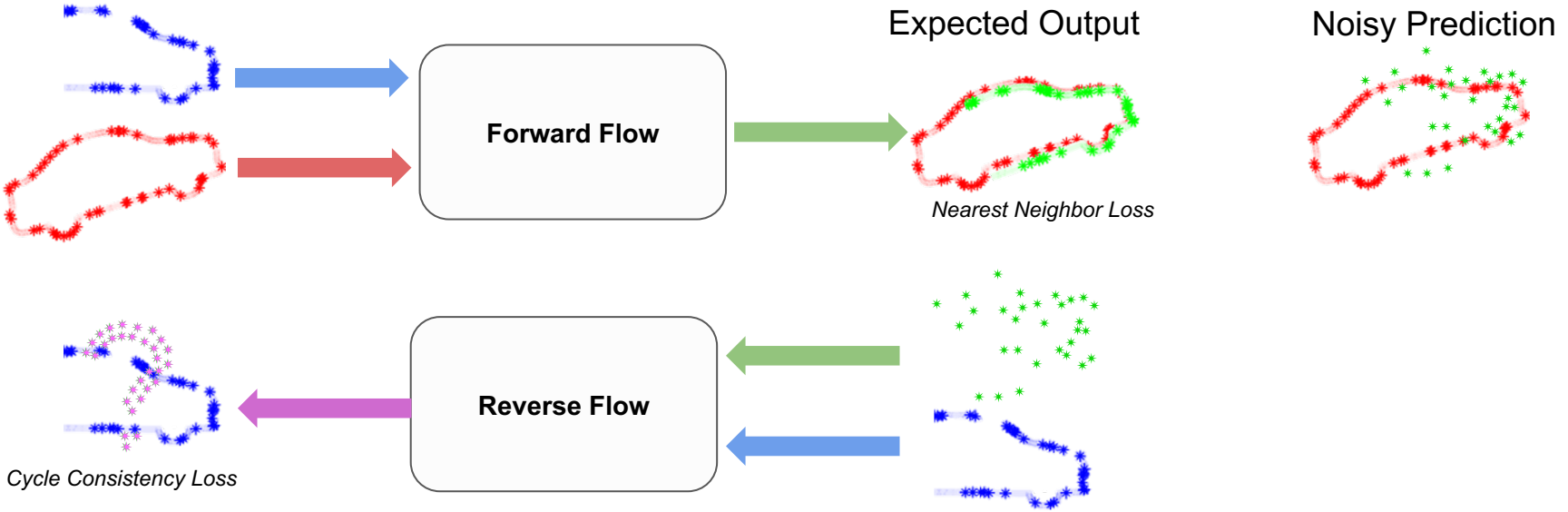
Self-Supervised Scene Flow



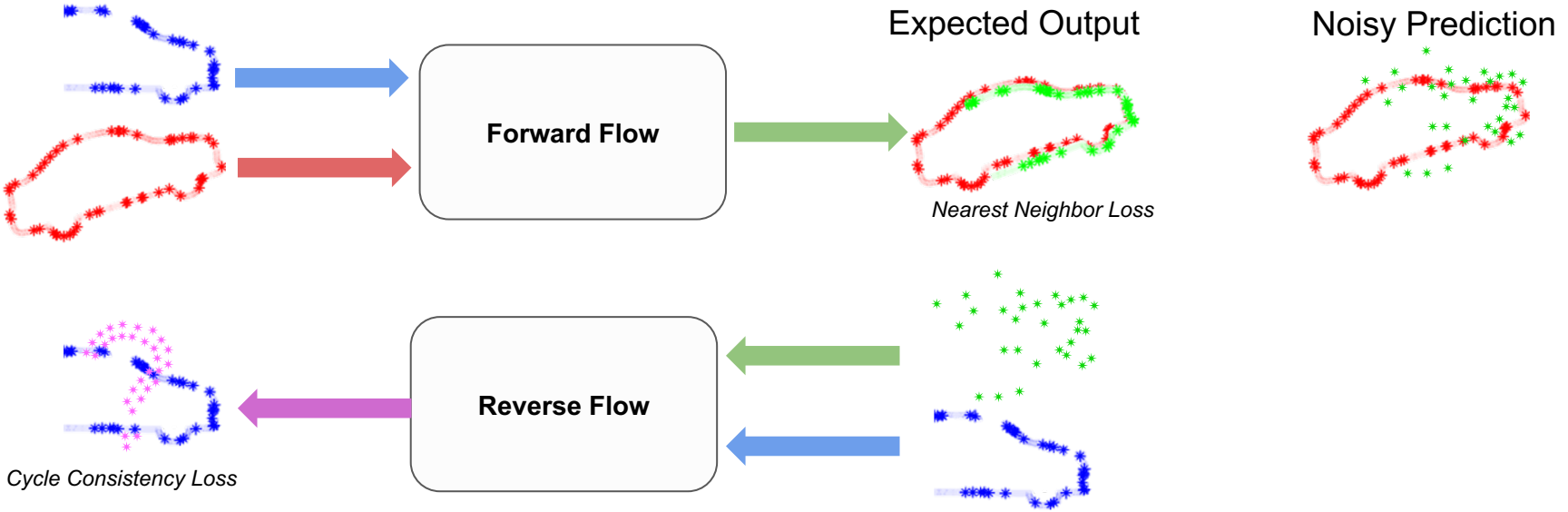
Self-Supervised Scene Flow



Self-Supervised Scene Flow

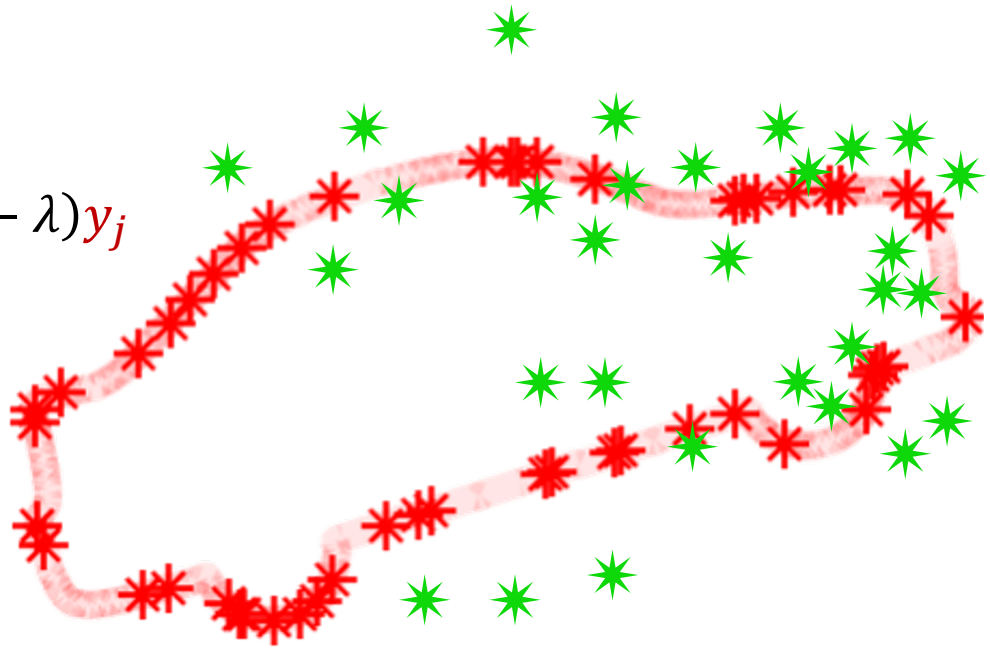


Self-Supervised Scene Flow



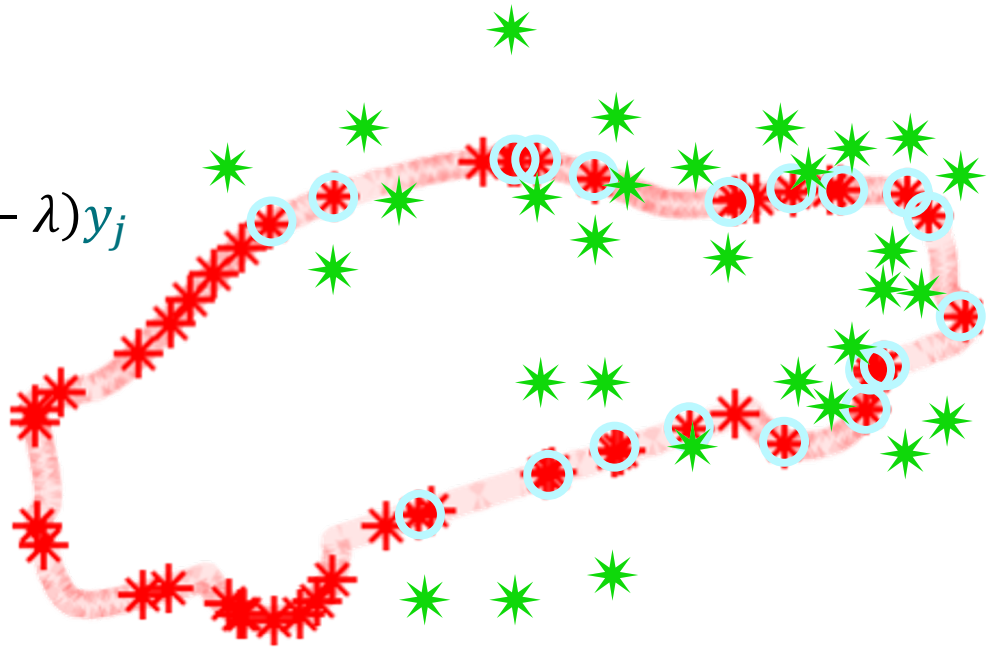
Anchored Predictions

$$\bar{x}_i' = \lambda \hat{x}_i' + (1 - \lambda) y_j$$



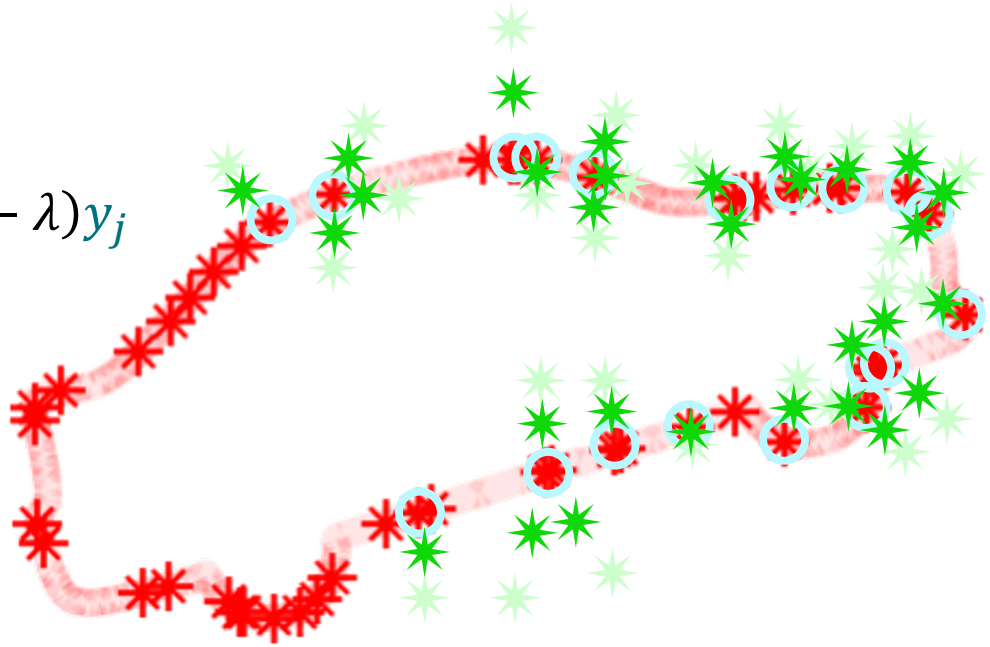
Anchored Predictions

$$\bar{x}_i' = \lambda \hat{x}_i' + (1 - \lambda) y_j$$



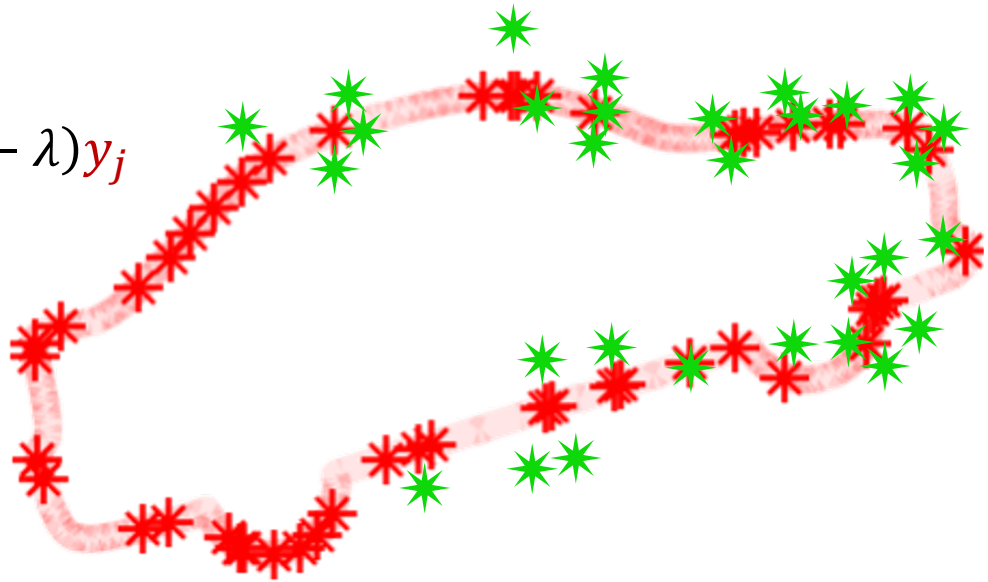
Anchored Predictions

$$\bar{x}_i' = \lambda \hat{x}_i' + (1 - \lambda) y_j$$

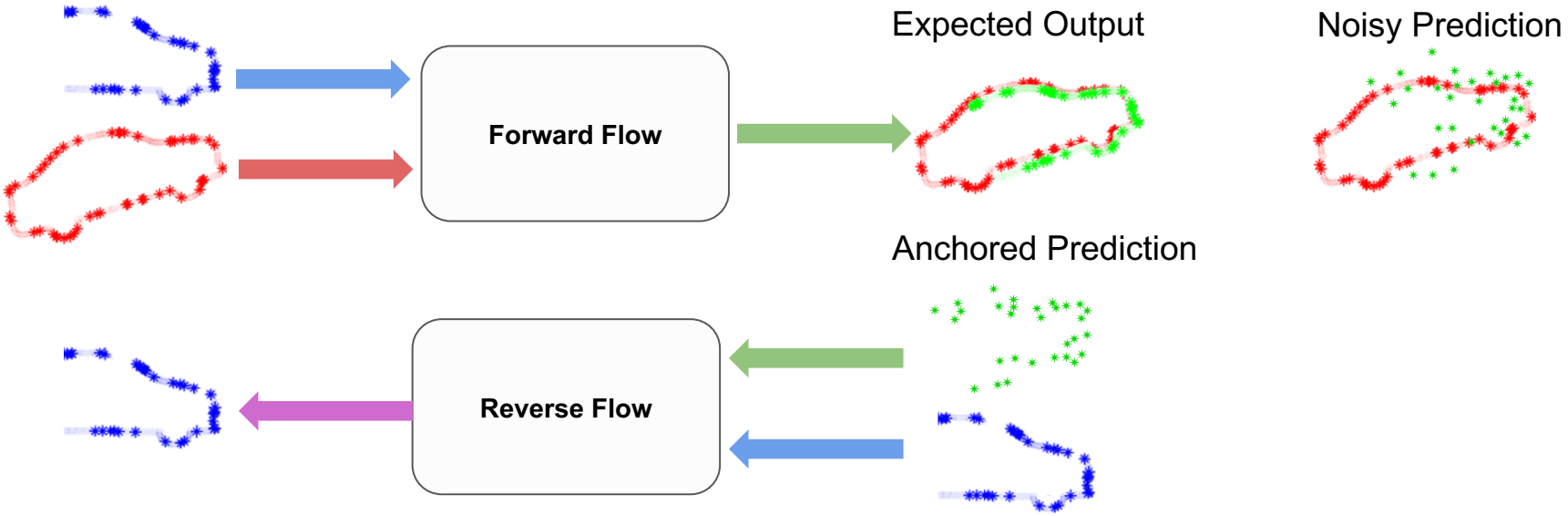


Anchored Predictions

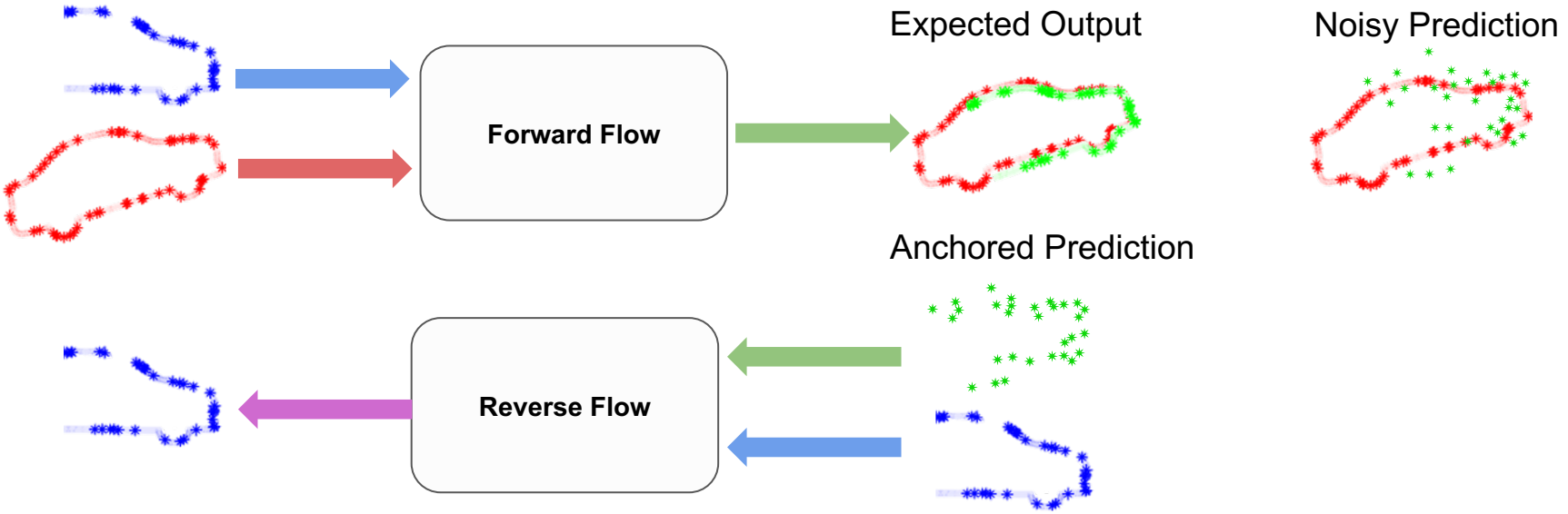
$$\bar{x}_i' = \lambda \hat{x}_i' + (1 - \lambda) y_j$$



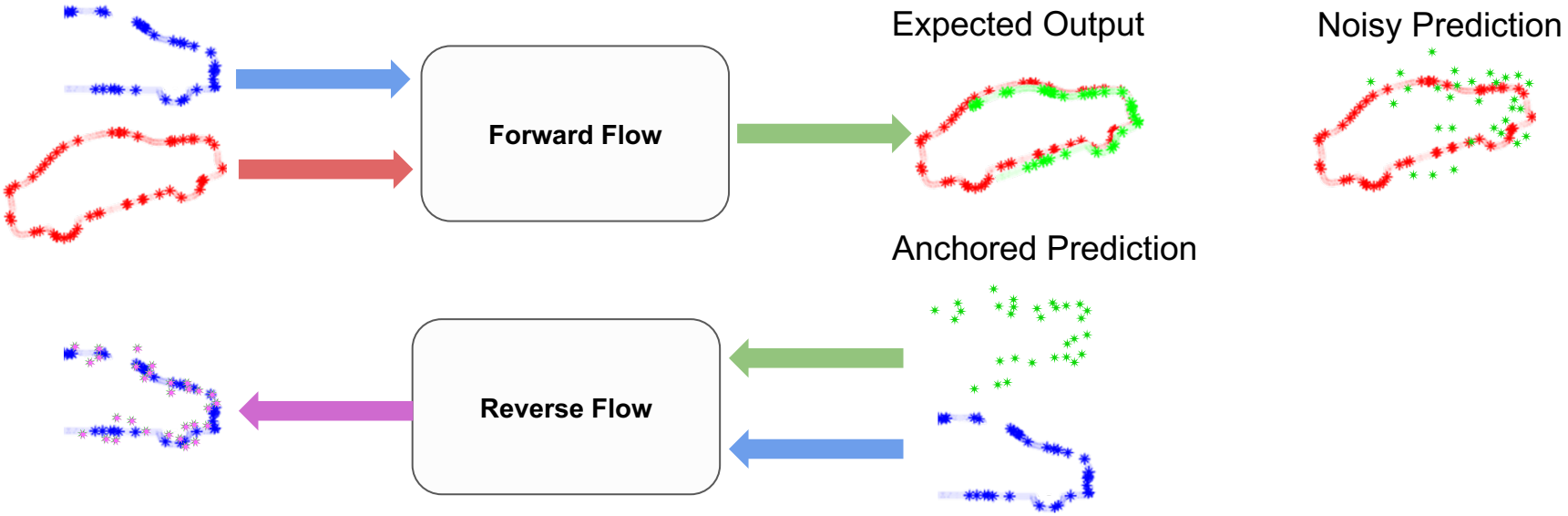
Self-Supervised Scene Flow



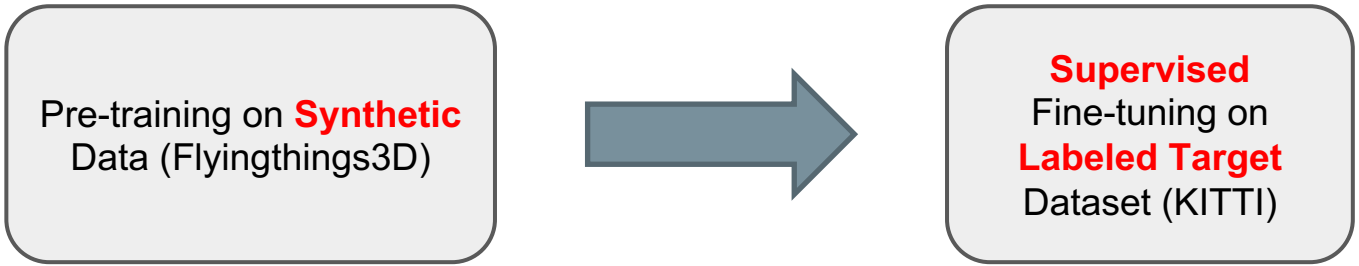
Self-Supervised Scene Flow



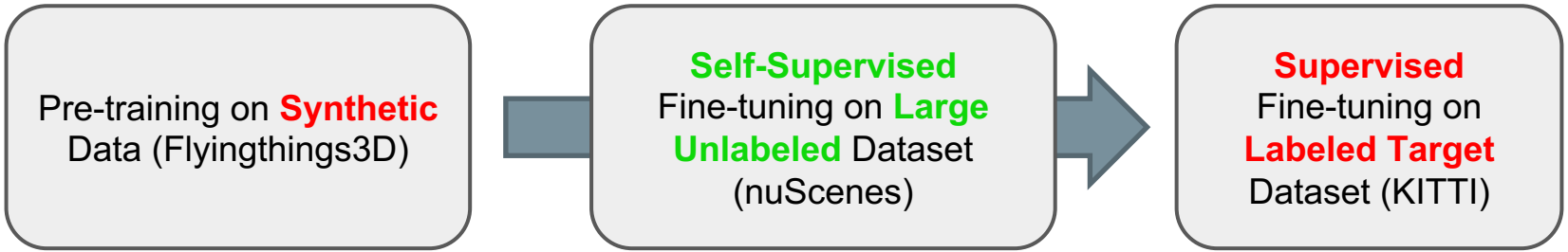
Self-Supervised Scene Flow



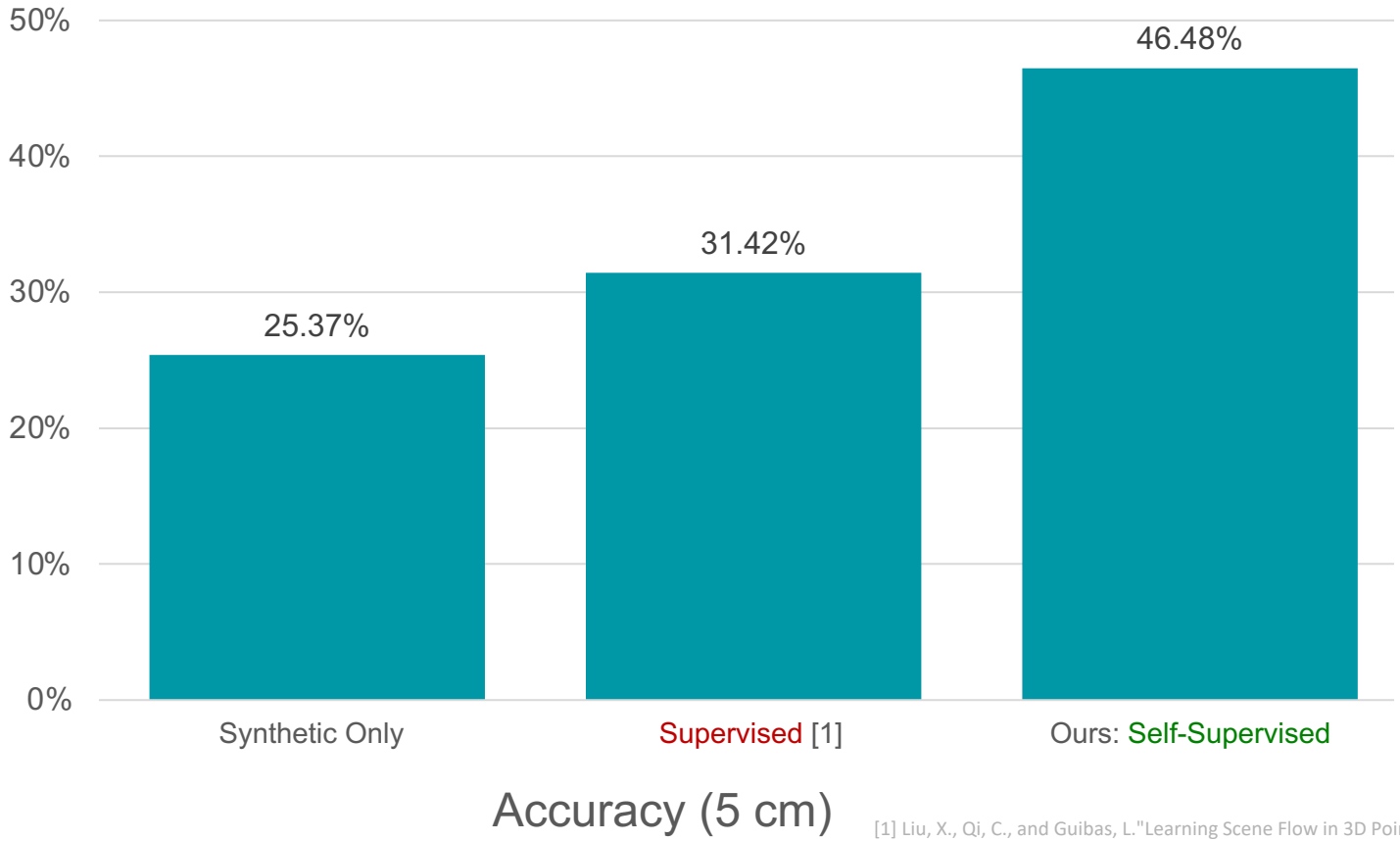
Training Procedure – Previous Work



Training Procedure – Our Work



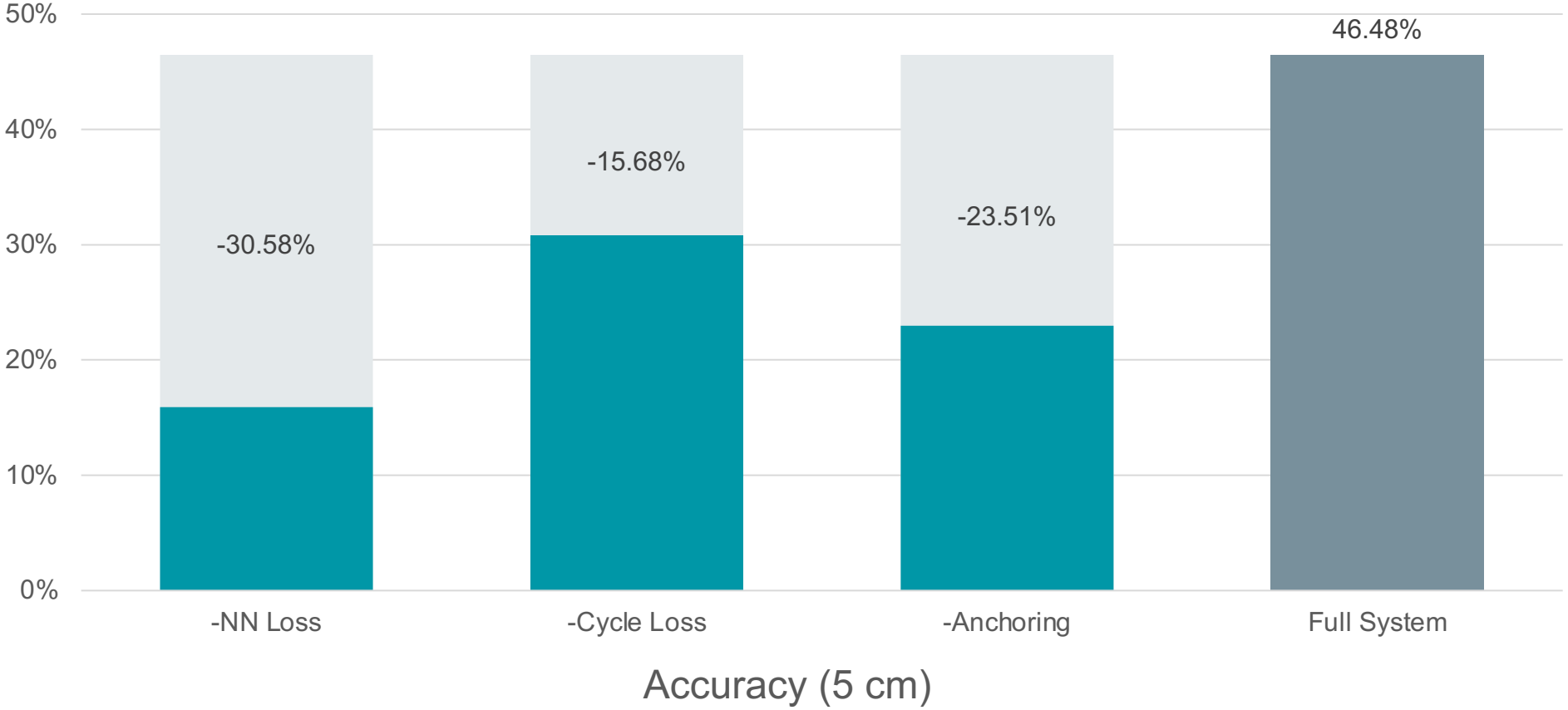
Results: Scene Flow Accuracy



[1] Liu, X., Qi, C., and Guibas, L."Learning Scene Flow in 3D Point Clouds." *CVPR* 2019.

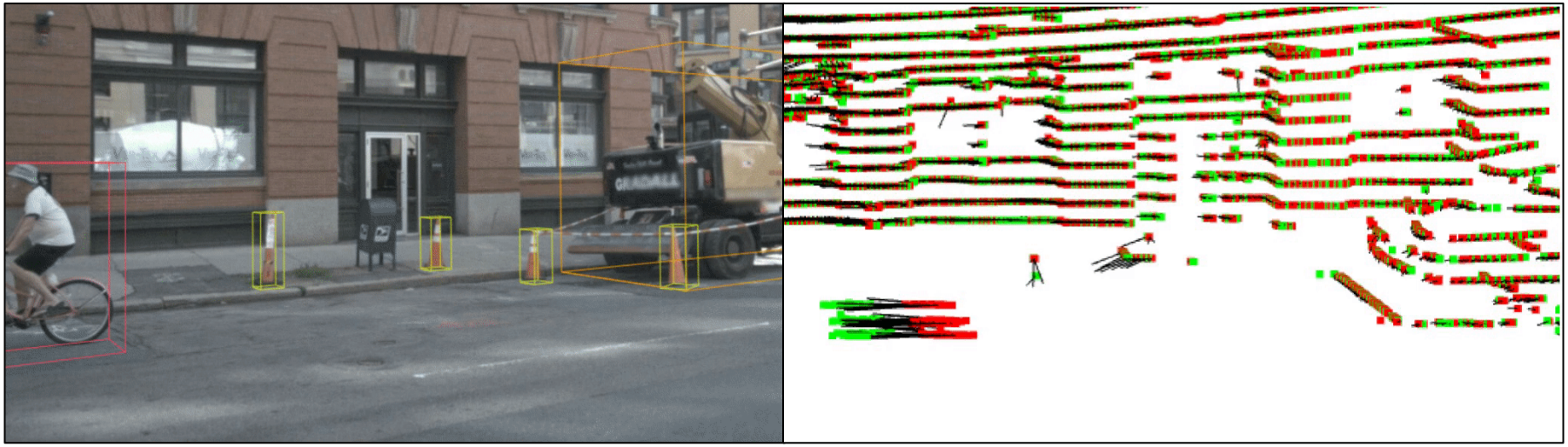
Ablation Analysis

nuScenes (Self-Supervised) + KITTI (Self-Supervised)



Qualitative Results: Tracking a Cyclist

Target Cloud (green) Source Cloud (red)

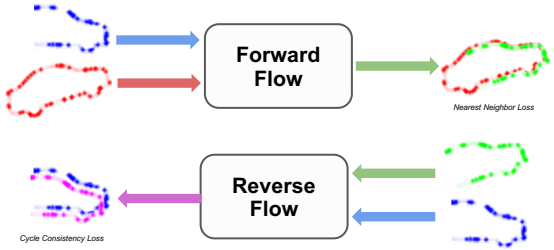


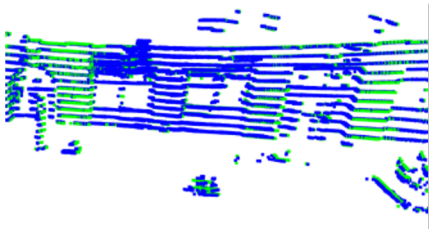
Conclusions

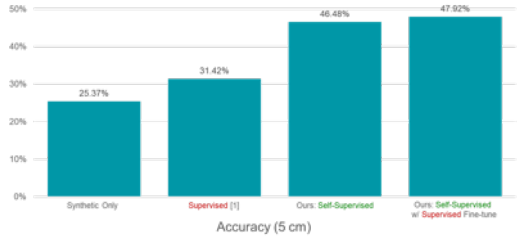
Self-supervised training method for scene flow

Leverage large unlabeled datasets

Exceed the performance of supervised training on a smaller labeled dataset





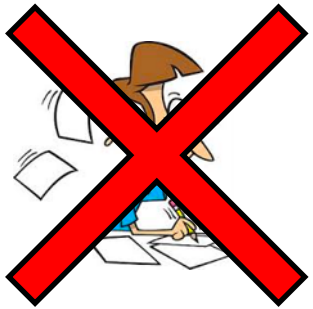


Self-Supervised Learning for Autonomous Driving

- Active Perception with Light Curtains (ECCV 2020)
- **Self-supervised 3D Scene Flow (CVPR 2020)**
- Self-supervised 3D Data Association (IROS 2020)

Himangi Mittal

Brian Okorn



Human labeling

Regular, easy driving

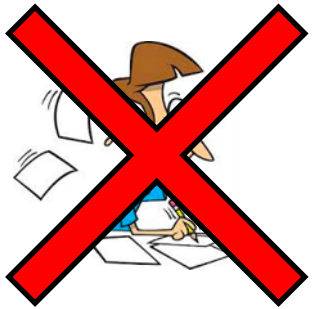
Difficult driving

Weird driving

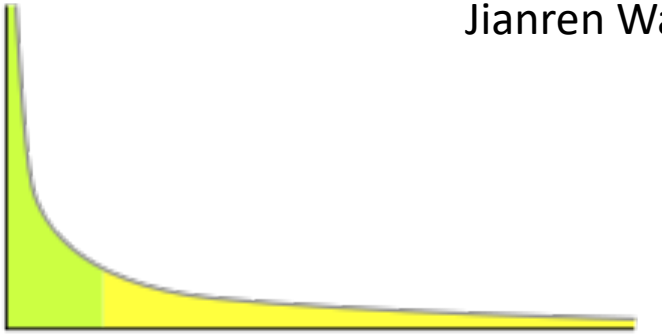
Self-Supervised Learning for Autonomous Driving

- Active Perception with Light Curtains (ECCV 2020)
- Self-supervised 3D Scene Flow (CVPR 2020)
- **Self-supervised 3D Data Association (IROS 2020)**

Jianren Wang



Human labeling



Regular, easy driving

Difficult driving

Weird driving

Traditional Tracking: Motion Prediction

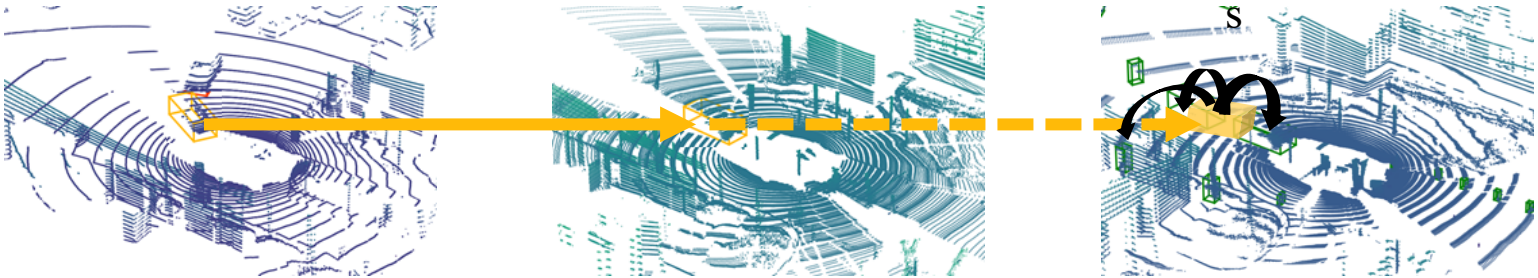
Kalman Filter

Frame 1, … , t -1

Track i

Frame t

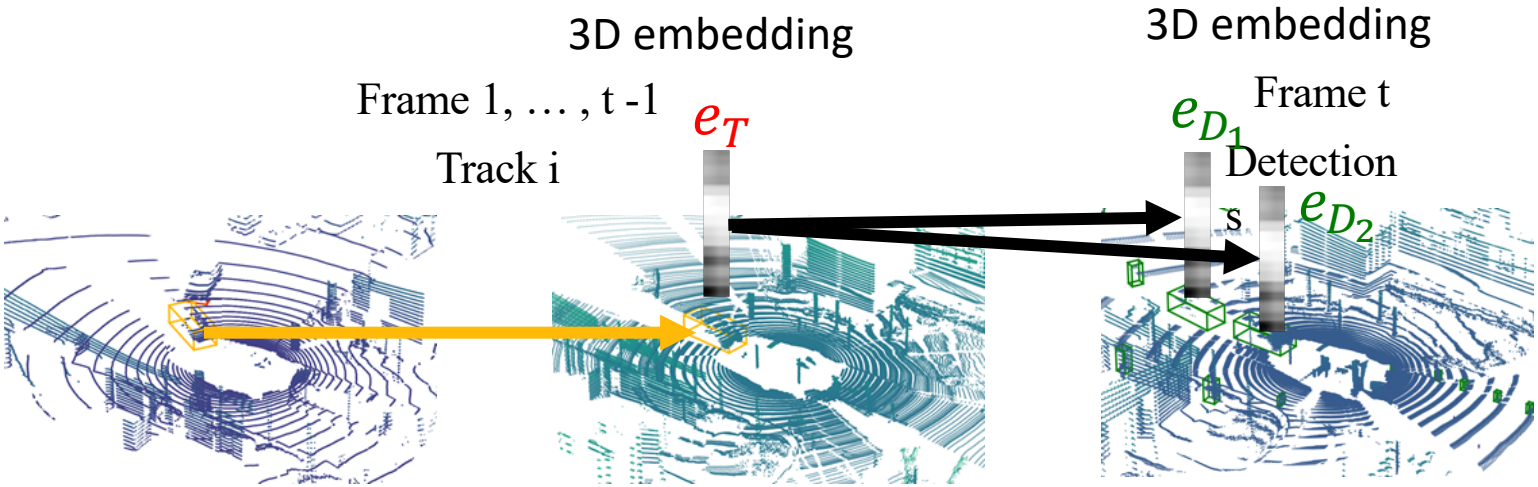
Detection



Motion distance

[1] Weng, Xinshuo, et al. “3D Multi-Object Tracking: A Baseline and New Evaluation Metrics”, IROS2020
[2] Zhu, Benjin, et al. “Probabilistic 3D Multi-Object Tracking for Autonomous Driving”, arXiv

Improved Tracking: Appearance Similarity



Appearance Similarity[1,2,3]: $\cos(e_T, e_D)$

Can we learn 3D embedding from unlabeled data?

[1] Giancola, Silvio, et al. “Leveraging Shape Completion for 3D Siamese Tracking.”, CVPR2019
[2] Weng, Xinshuo, et al. “Graph Neural Network for 3D Multi-Object Tracking with 2D-3D Multi-Feature Learning.”, CVPR2019
[3] Zhang, Weiwei, et al. “Robust Multi-Modality Multi-Object Tracking.”, ICCV2019

e_{D_1}

Method - Overview

Sensor Data

↓

Object Detection

↓

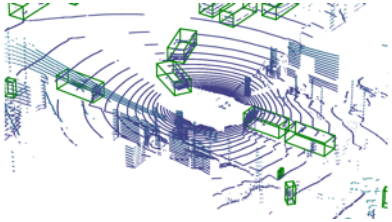
Motion-based Data Association

↓

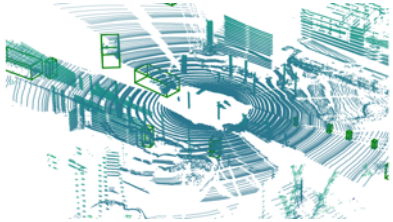
Uncertainty-aware Weighting

↓

Learning 3D Embedding



Frame t



Frame t + k

Method - Overview

Sensor Data

↓

Object Detection

↓

Motion-based Data Association

↓

Uncertainty-aware Weighting

↓

Learning 3D Embedding

Frame t

Frame t + k

Method - Overview

Sensor Data

↓

Object Detection

↓

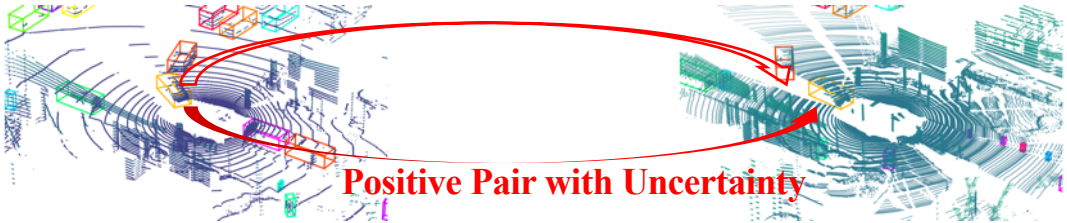
Motion-based Data Association

↓

Uncertainty-aware Weighting

↓

Learning 3D Embedding



Frame t

Frame t + k

Method - Overview

Sensor Data

↓

Object Detection

↓

Motion-based Data Association

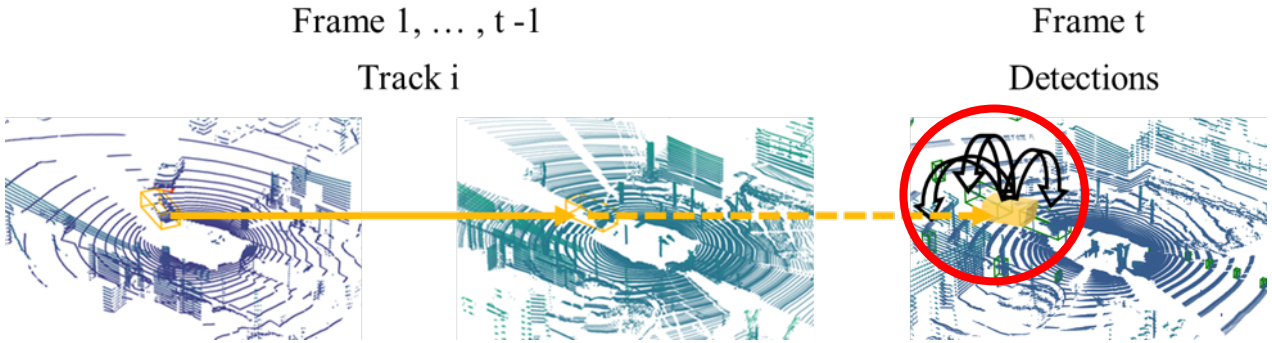
↓

Uncertainty-aware Weighting

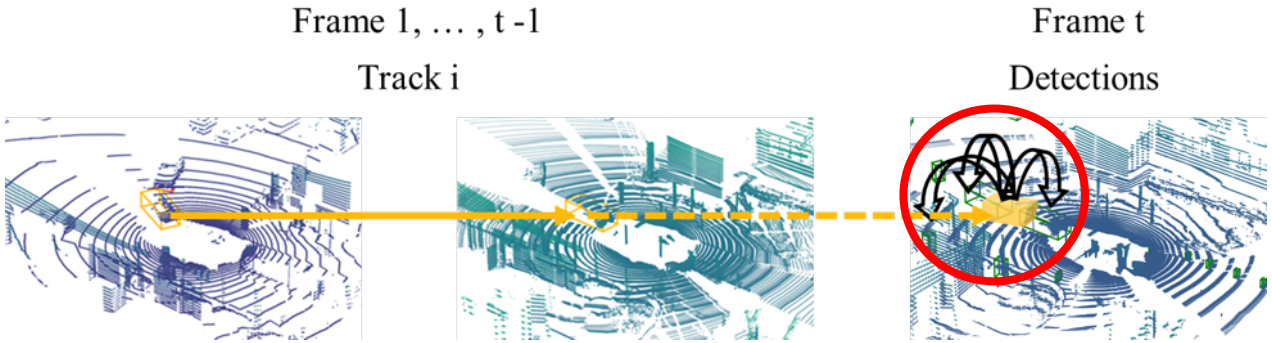
↓

Learning 3D Embedding

Obtain Possible Associations by Kalman Filter Tracking



Weight Associations by Uncertainty



		Detections		
		1	2	3
Tracks	1	67	37	34
	2	44	6	18
	3	89	17	32

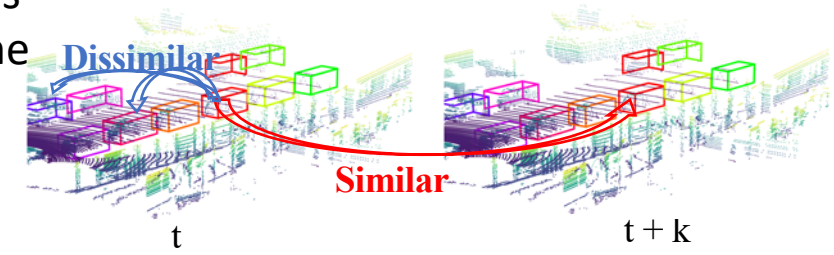
$$Association\ Uncertainty(AU) = 1 - \exp(-\min\left(\frac{17}{6}, \frac{18}{6}\right))$$

$$Cumulative\ Association\ Uncertainty(CAU) = \prod_{i=t}^{t+k} AU(t)$$

$$weighted\ loss = loss \times CAU(t, t+k)$$

Metric Learning via Triplet Loss

Negative pairs from the same frame



Positive pairs from motion-based tracking

- Triplet loss:

$$L\left(e_a^t, e_p^{t+k}, e_n^t\right) = \max(\cos(e_a^t, e_n^t) - \cos\left(e_a^t, e_p^{t+k}\right) + M, 0)$$

$$weighted\ loss = loss \times CAU(t, t+k)$$

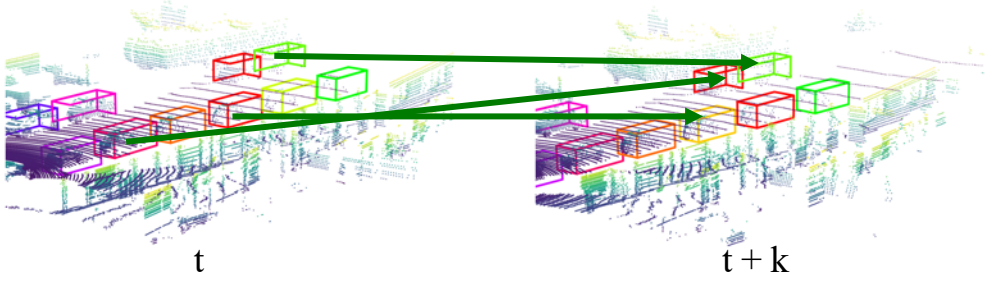
e_a^t: anchor detection in frame t e_p^{t+k} : same object in frame t + k e_n^t: different object in frame t M: Margin

Combining Cues

Logistic regression combining

- Mahalanobis Distance (M)
- Detection Score (D)
- Cosine distance with appearance embedding (A)

This part is fully supervised but does not require much data



Experiment Setups

Single Object Tracking

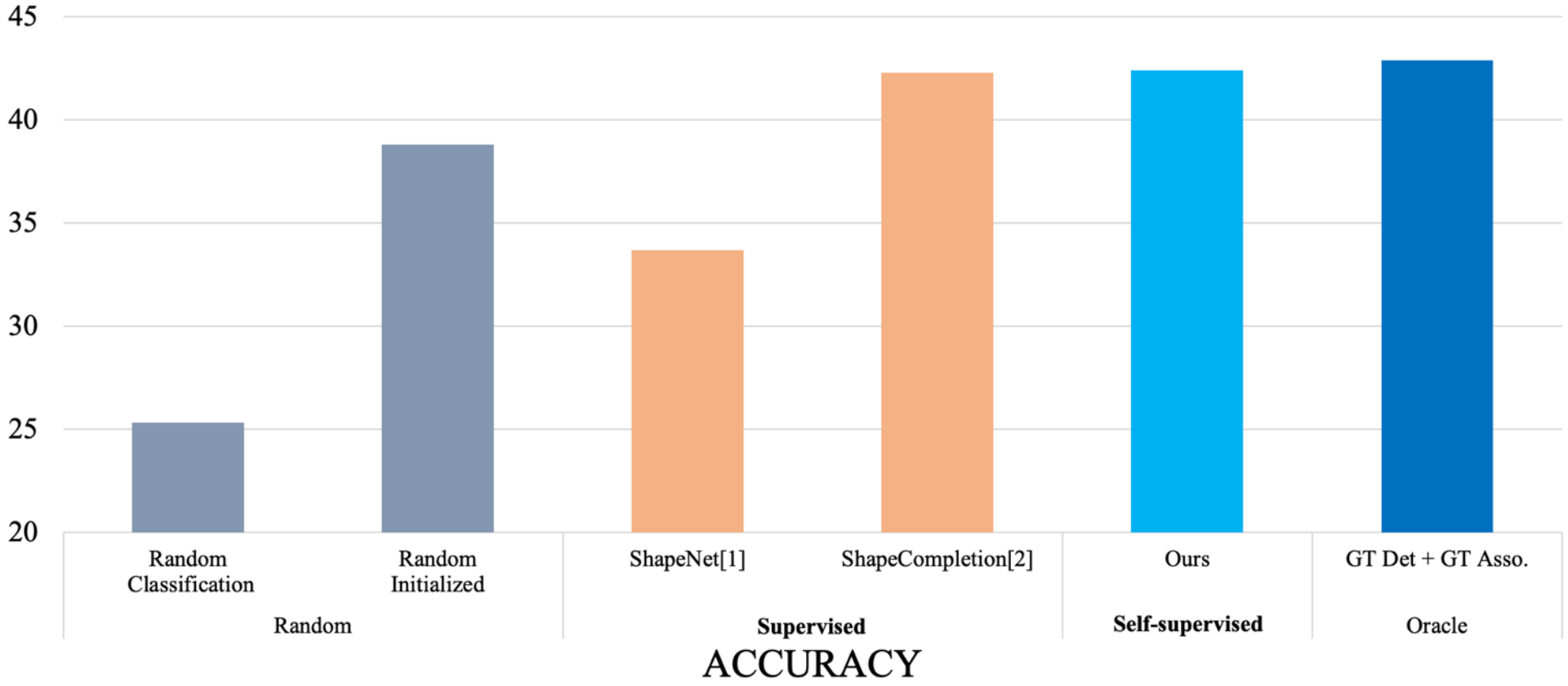
- Embedding train with NuScenes[1]
- Evaluate with KITTI[2]
- Evaluation Metric: Classification Accuracy

Multi-Object Tracking

- Embedding train with NuScenes[1]
- Logistic regression train with NuScenes[1]
- Evaluate with NuScenes[1]
- Evaluation Metric: AMOTA

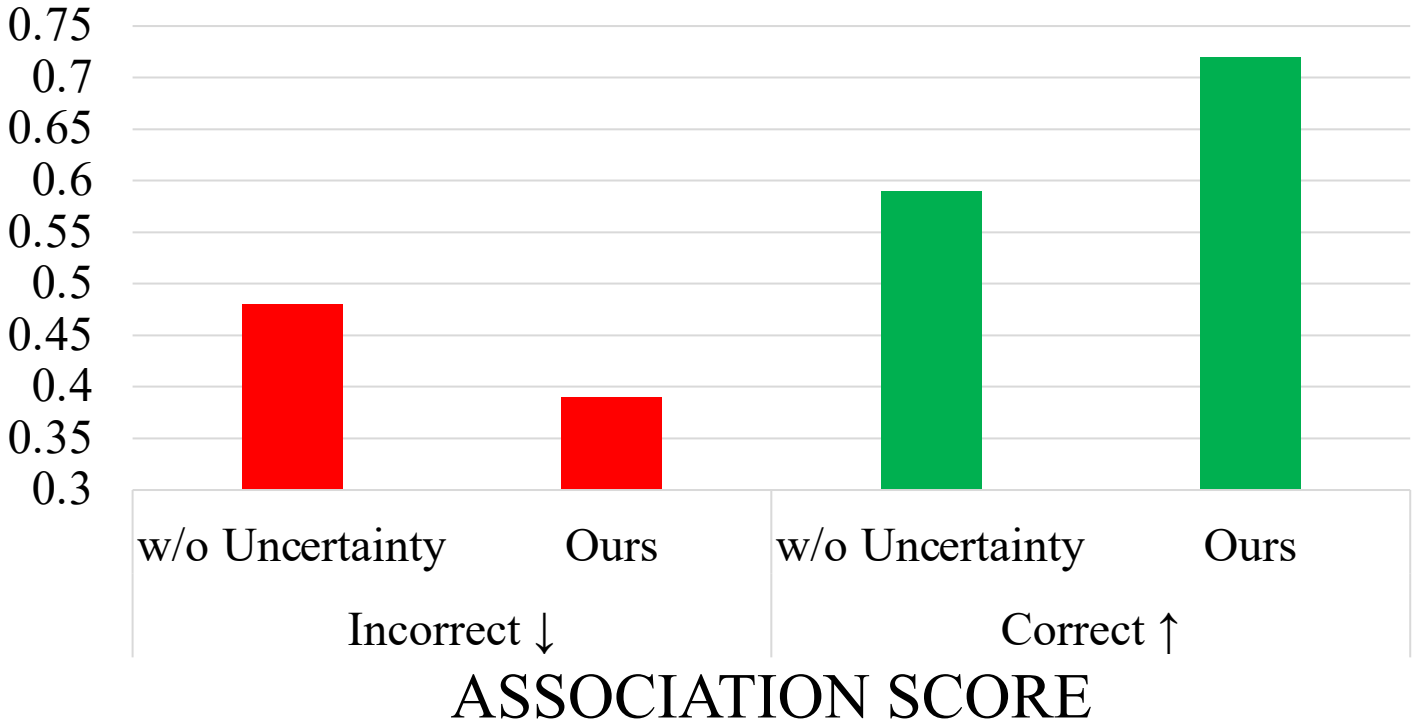
[1] Caesar, Holger et al. “nuScenes: A multimodal dataset for autonomous driving.”, CVPR2020
[2] Geiger, Andreas et al. “Vision meets Robotics: The KITTI Dataset.”, IJRR2013

Experiments – Single Object Tracking



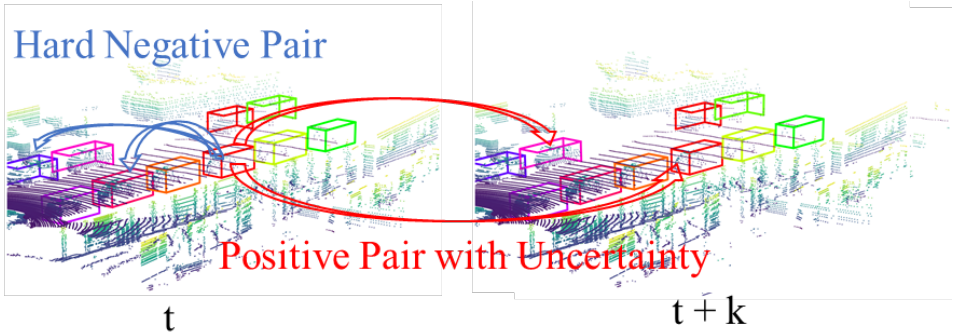
[1] Chang, Angel, et al. “ShapeNet: An Information-Rich 3D Model Repository”, arXiv
[2] Giancola, Silvio, et al. “Leveraging Shape Completion for 3D Siamese Tracking.”, CVPR2019

Experiments – Uncertainty



Conclusion

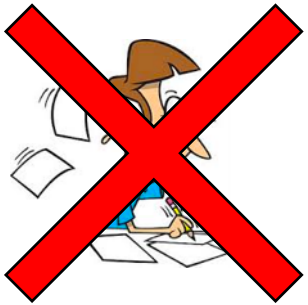
- **Self-supervised** representation learning method for 3D data association
- We incorporate **tracking uncertainty** for robust self-supervised learning to handle tracking errors
- Avoids the need for extensive manual annotation; our self-supervised learning approach can make use of large **unlabeled datasets**



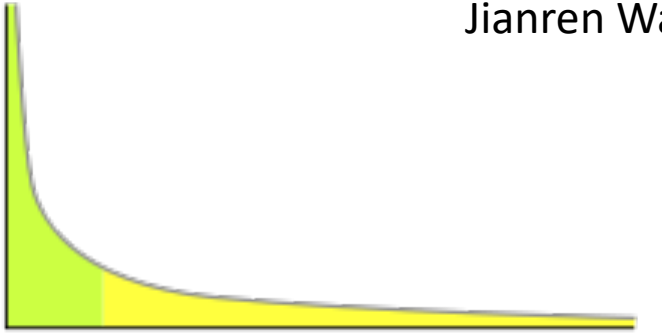
Self-Supervised Learning for Autonomous Driving

- Active Perception with Light Curtains (ECCV 2020)
- Self-supervised 3D Scene Flow (CVPR 2020)
- **Self-supervised 3D Data Association (IROS 2020)**

Jianren Wang



Human labeling



Regular, easy driving

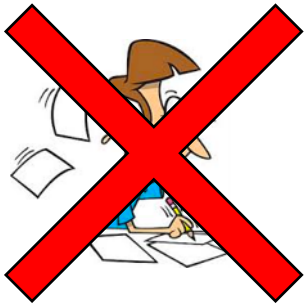
Difficult driving

Weird driving

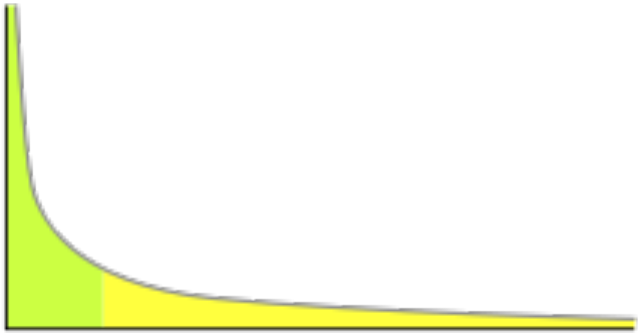
Self-Supervised Learning for Autonomous Driving

- Active Perception with Light Curtains (ECCV 2020)
- Self-supervised 3D Scene Flow (CVPR 2020)
- Self-supervised 3D Data Association (IROS 2020)

Questions?



Human labeling



Regular, easy driving

Difficult driving

Weird driving