Sequential Forecasting of 100k Points

Uber ATG Toronto Reading Group, Presented by Andrei Bârsan

April 15, 2020

Authors: Xinshuo Weng, Jianren Wang, Sergey Levine, Kris Kitani, Nicholas Rhinehart

Paper: https://arxiv.org/abs/2003.08376

Uber

Agenda

01 Overview & Motivation
02 Why Is Unsupervised Prediction Hard?
03 Proposed Method
04 Results & Analysis
05 Insights & Conclusion

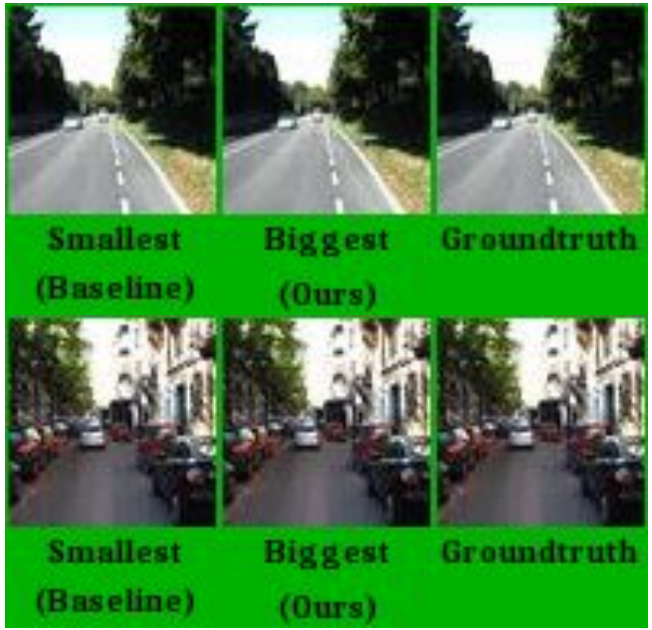
Please feel free to stop me at any time if you have questions!

Overview & Motivation

- Predicting the future is important but challenging
- In principle, we have **infinite** training data
- Leveraging these “free” labels is very challenging
- This talk: **LiDAR point cloud forecasting**

Why is Unsupervised Prediction Hard?

- Multimodality
- Difficulties learning dynamics and “common sense” from scratch
- For camera videos
  - SotA limited to 1--2 seconds
- Can we do better with LiDAR?



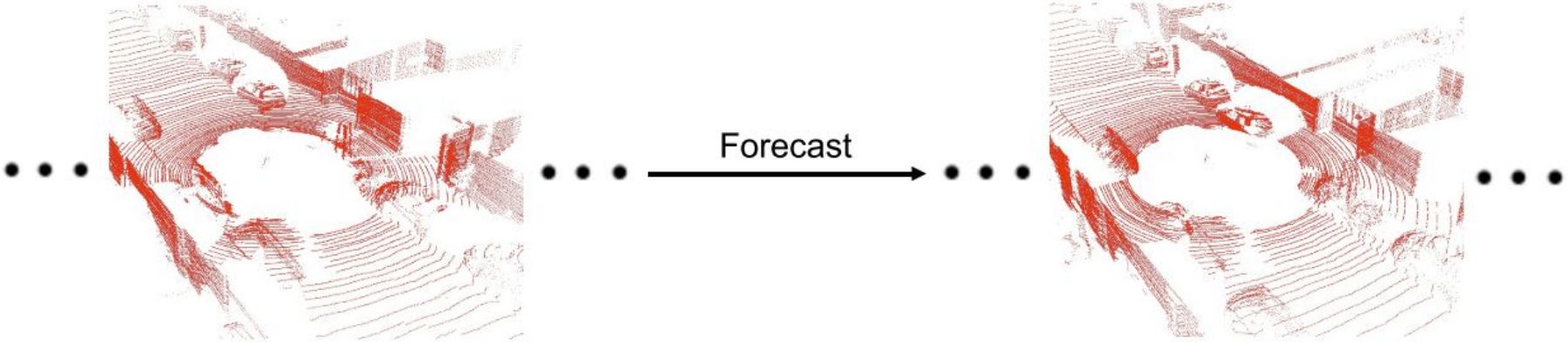
Villegas, Ruben, et al. "High fidelity video prediction with large stochastic recurrent neural networks." *Advances in Neural Information Processing Systems*. 2019.

Proposed Contributions

1. Define a new task, *Scene Point Cloud Sequence Forecasting*
2. Present a simple & effective method for this task
3. Present a prediction method based on this
4. Present new metrics to overcome limitations of existing ones (ADE/FDE vs. recall trade-off)

Proposed Method

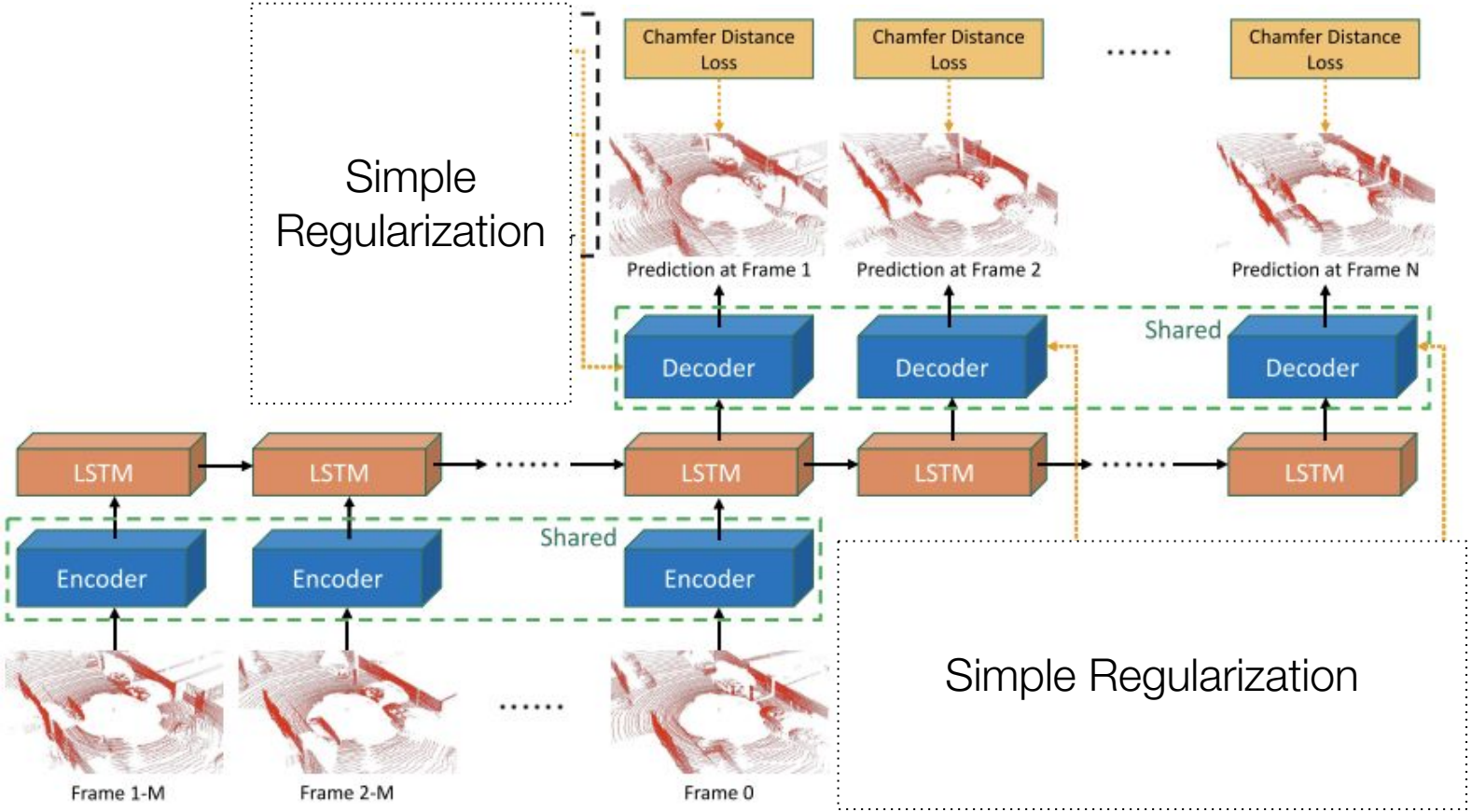
Scene Point Cloud Sequence Forecasting (Ours)



Scene point cloud in past M frames

Scene point cloud in future N frames

Proposed Method: Architecture



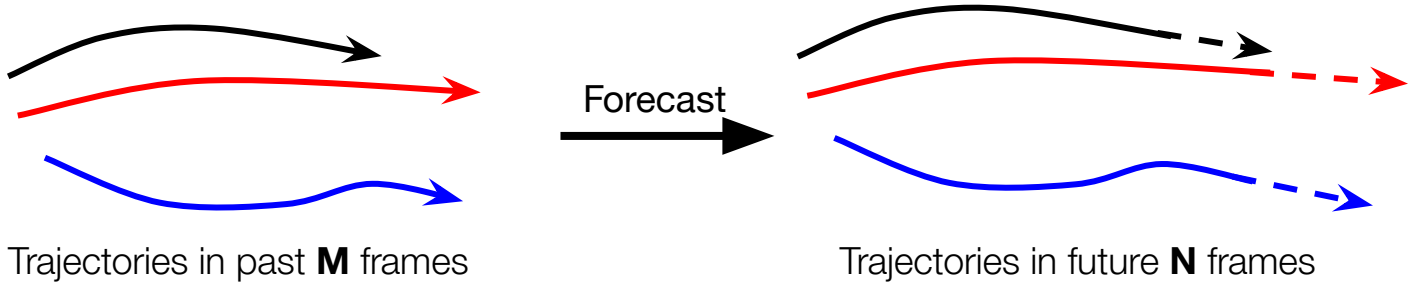
Losses for Sensor Forecasting

- Range images
  - L1-loss on range images
  - BCE on sparsity mask
  - Chamfer Distance Loss on **unpacked** point cloud
- 3D points
  - Chamfer Distance Loss on **decoded** point cloud

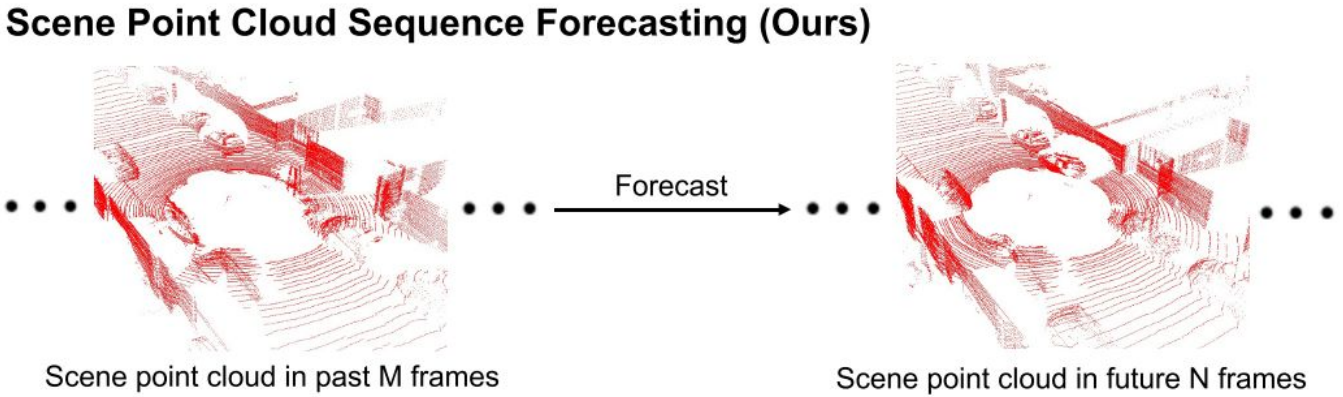
$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2$$

Prediction as Tracking

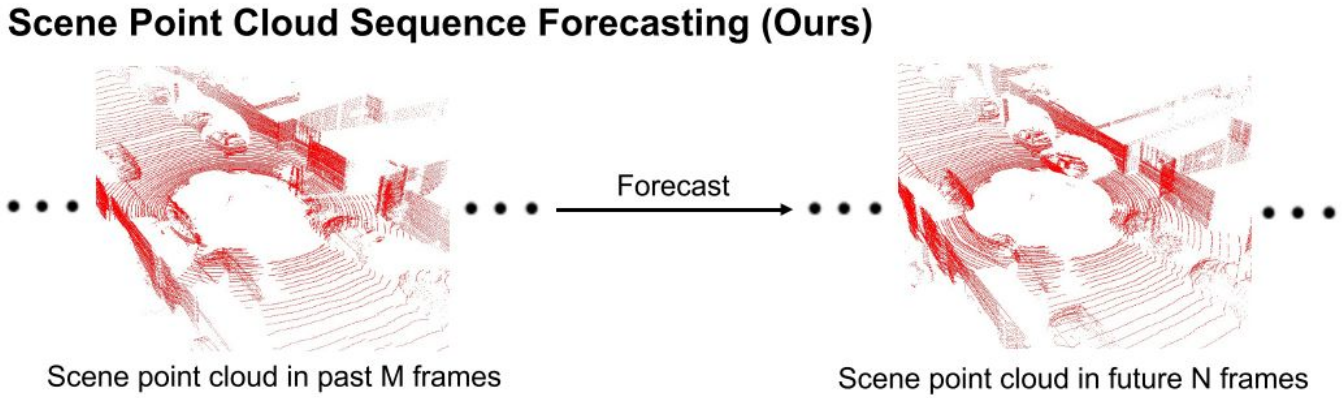
Standard Approach



Proposed Approach

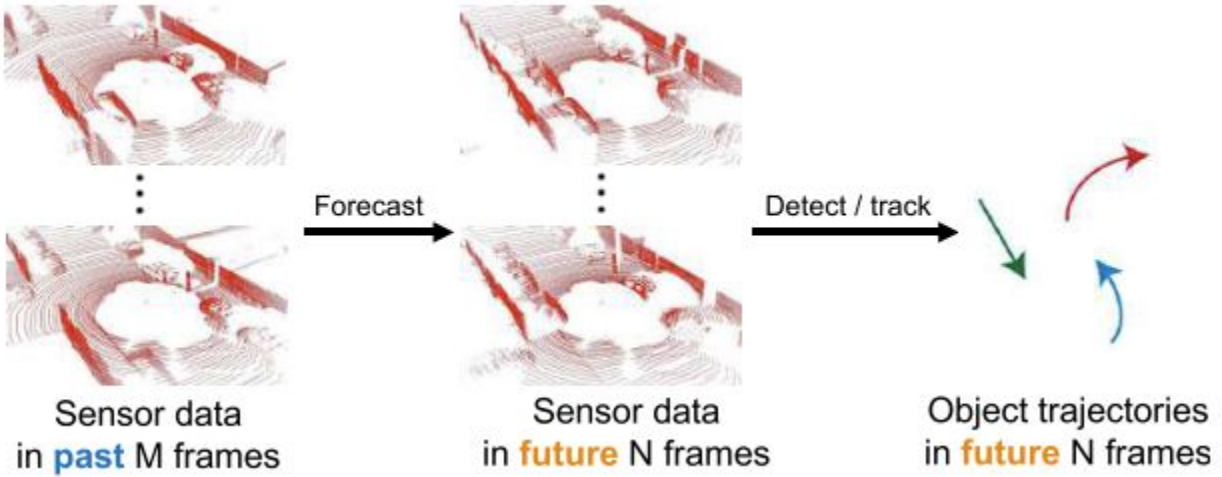


Prediction as Tracking



- Notes
  - No need for **tracking** labels
  - Still need a trained detector! (Point R-CNN is used.)

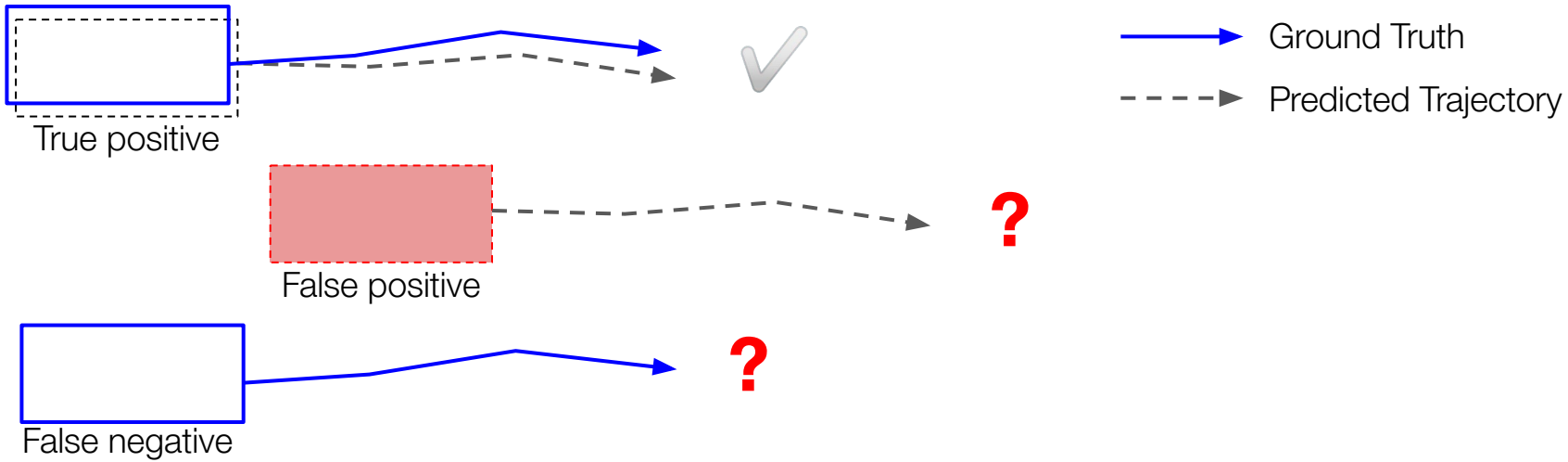
Prediction as Tracking



- Sensor forecasting done with their method
- Detection done with Point R-CNN
- Tracking done with a Kalman Filter-based tracker

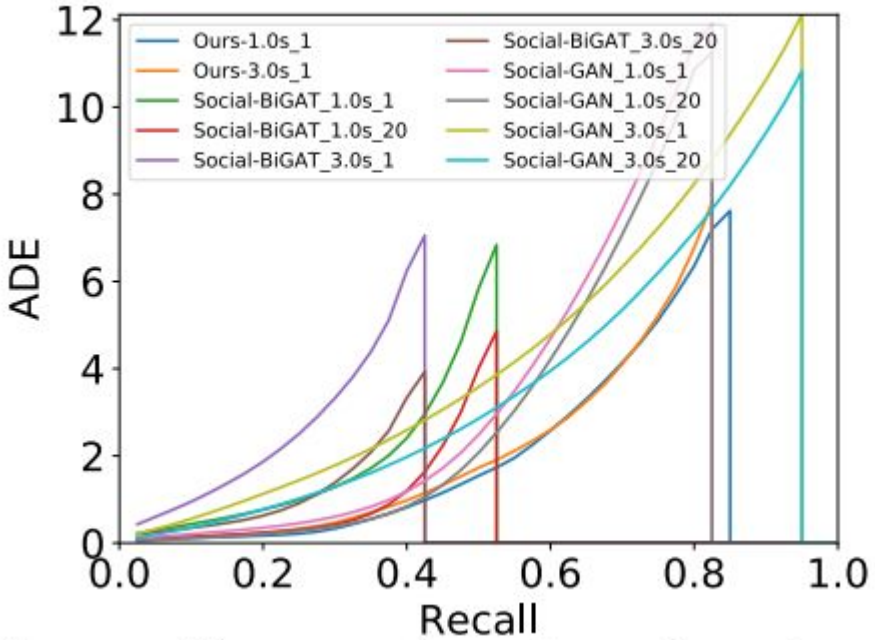
Prediction Metric Challenges

- Evaluating predictions from **GT** detection = easy, 1:1 mapping
- Evaluating predictions from **real** detection = hard, not a 1:1 mapping

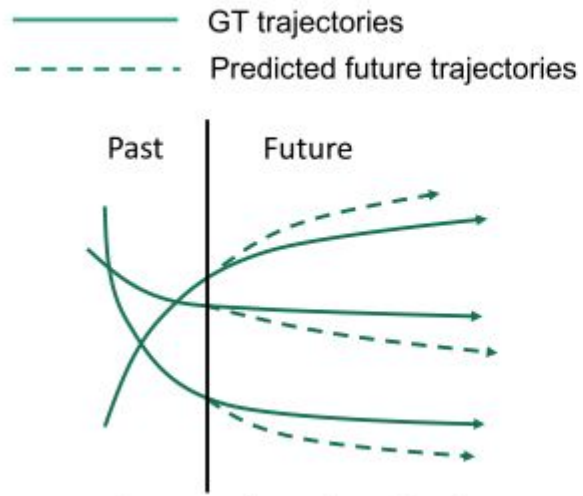


Prediction Metric Challenges

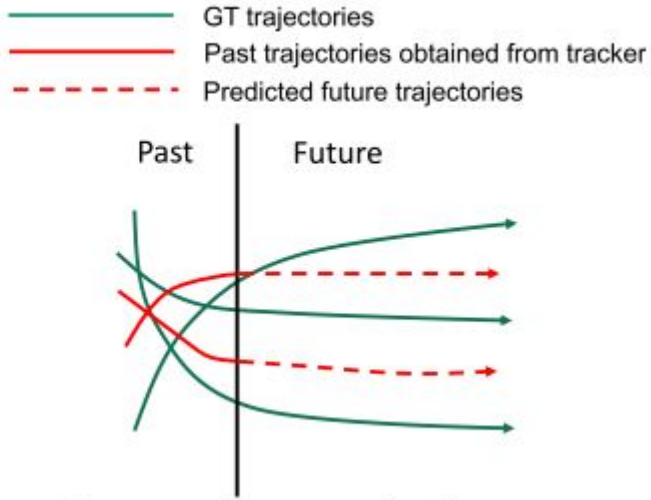
- Average Displacement Error vs. trajectory recall
- **Issue:** Comparing ADE @ X recall doesn't paint a full picture
- **Solution:** Take the integral
- Alternatives:
  - Just show the curve
  - System-level metrics



New Prediction Metrics: Idea



Conventional evaluation

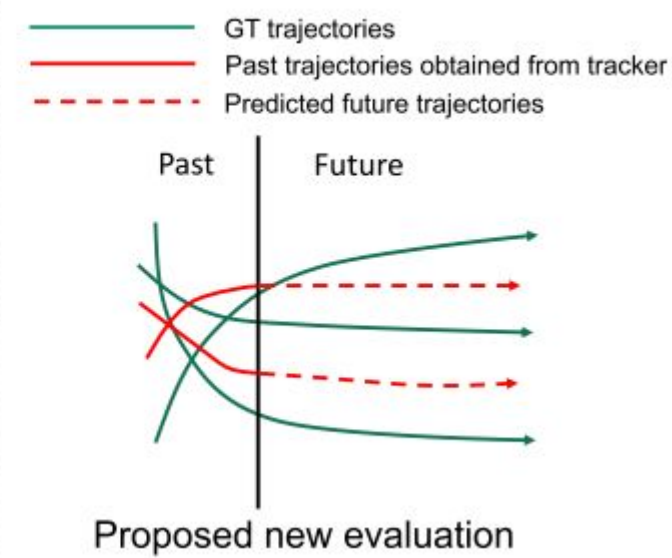


Proposed new evaluation

New Prediction Metrics: Idea

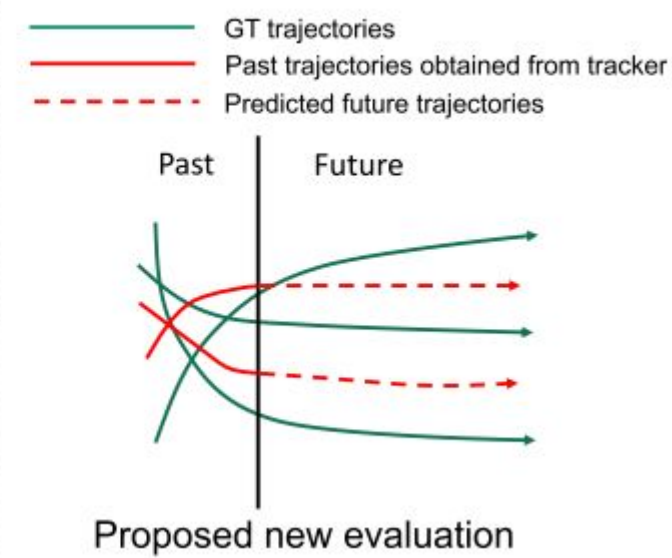
- Measure
  - Average ADE (AADE)
  - Average FDE (AFDE)
- Implemented as discrete sum

$$\frac{1}{|R|}\sum_{r\in R}(ADE \text{ @ recall } r)$$



New Prediction Metrics: GT Association

- Predicted trajectories matched with GT using the Hungarian algorithm
- In contrast to, e.g., IoU-based association
- May result in overly optimistic metrics



Results

- Sensor Forecasting
- BEV Tracking from LiDAR
- Datasets
  - KITTI, nuScenes

Results: Sensor Forecasting

- Compare predicted vs. true point cloud
  - Chamfer Distance
  - Earth-Mover Distance

Earth Mover's distance Consider $S_1, S_2 \subseteq \mathbb{R}^3$ of equal size $s = |S_1| = |S_2|$. The EMD between A and B is defined as:

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

where $\phi : S_1 \to S_2$ is a bijection.

Approximated for computational reasons (more info)

Results: Sensor Forecasting

Deep Closest Point

Point Clouds + PointNet

Range Images + 2D CNN

Datasets	Metrics	Identity	GT-Ego	Est-Ego	Align-ICP	Align-[63]	SceneFlow	**Ours+Point**	**Ours+RM**
KITTI-1.0s	CD↓	12.82	5.47	9.18	6.13	6.02	3.15	1.71	**0.89**
	EMD↓	526.87	391.03	495.21	418.25	439.17	291.63	211.47	**128.81**
KITTI-3.0s	CD↓	13.31	7.91	11.31	9.14	9.57	5.08	1.95	**0.94**
	EMD↓	602.89	452.81	502.83	470.25	493.26	351.46	267.42	**175.54**
NuScenes-1.0s	CD↓	8.42	2.16	4.91	4.04	3.50	1.93	1.03	**0.35**
	EMD↓	461.63	168.37	299.13	281.53	270.81	117.41	135.94	**78.37**
NuScenes-3.0s	CD↓	10.16	2.85	6.52	7.13	5.27	3.81	1.37	**0.41**
	EMD↓	494.81	190.14	370.91	419.37	332.97	294.53	128.26	**91.83**

Results: Tracking

Originally for Pedestrians

Originally for Pedestrians

Datasets	Metrics	Samples	Conv-Social [11]	Social-GAN [21]	Social-BiGAT [32]	TraPHic [8]	**Ours**
KITTI-1.0s	AADE↓	1	0.792	0.524	1.099	0.470	**0.317**
		20	0.623	0.340	0.443	0.382	–
	AFDE↓	1	1.285	0.886	1.708	0.889	**0.405**
		20	1.152	0.511	0.546	0.613	–
KITTI-3.0s	AADE↓	1	1.692	1.362	2.720	1.432	**0.408**
		20	1.593	0.984	1.231	0.725	–
	AFDE↓	1	2.670	2.267	3.938	2.536	**0.504**
		20	2.385	1.512	1.405	1.118	–

Ablation Study: Amount of Data

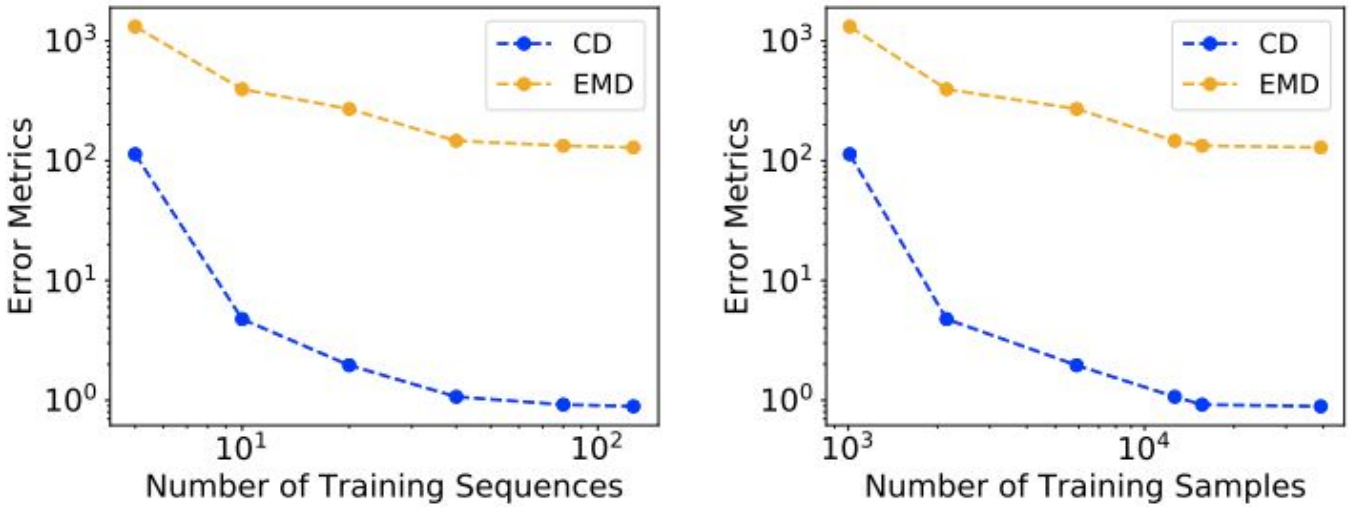
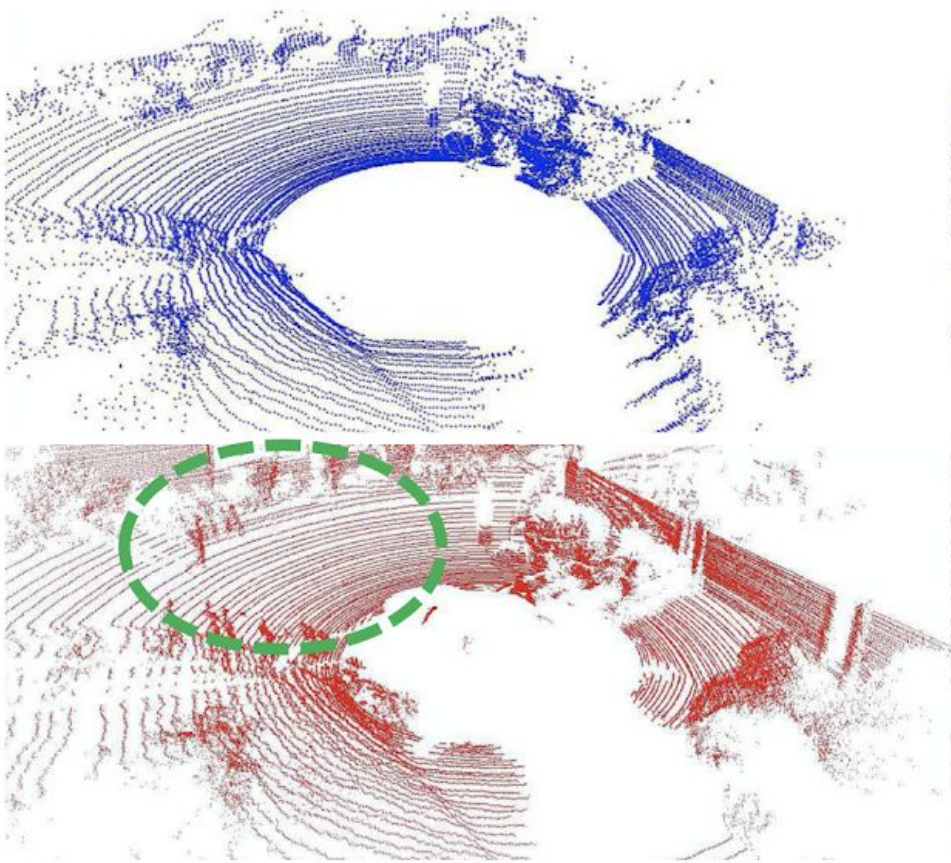
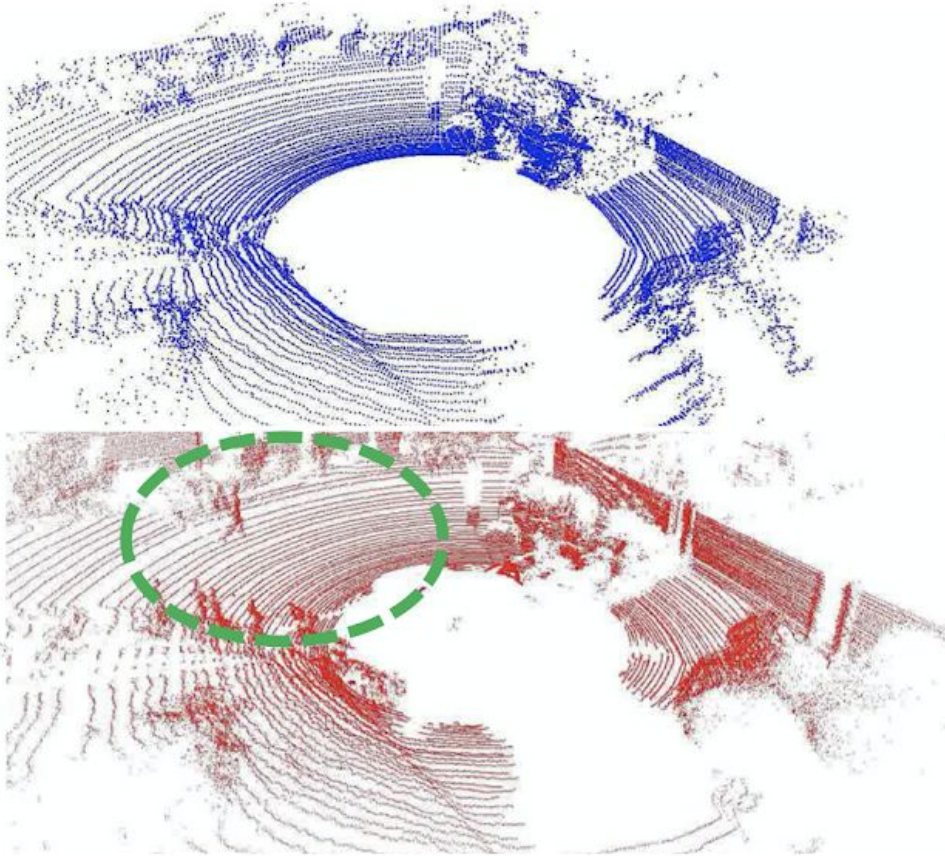


Fig. 9: Error vs. amount of training data.

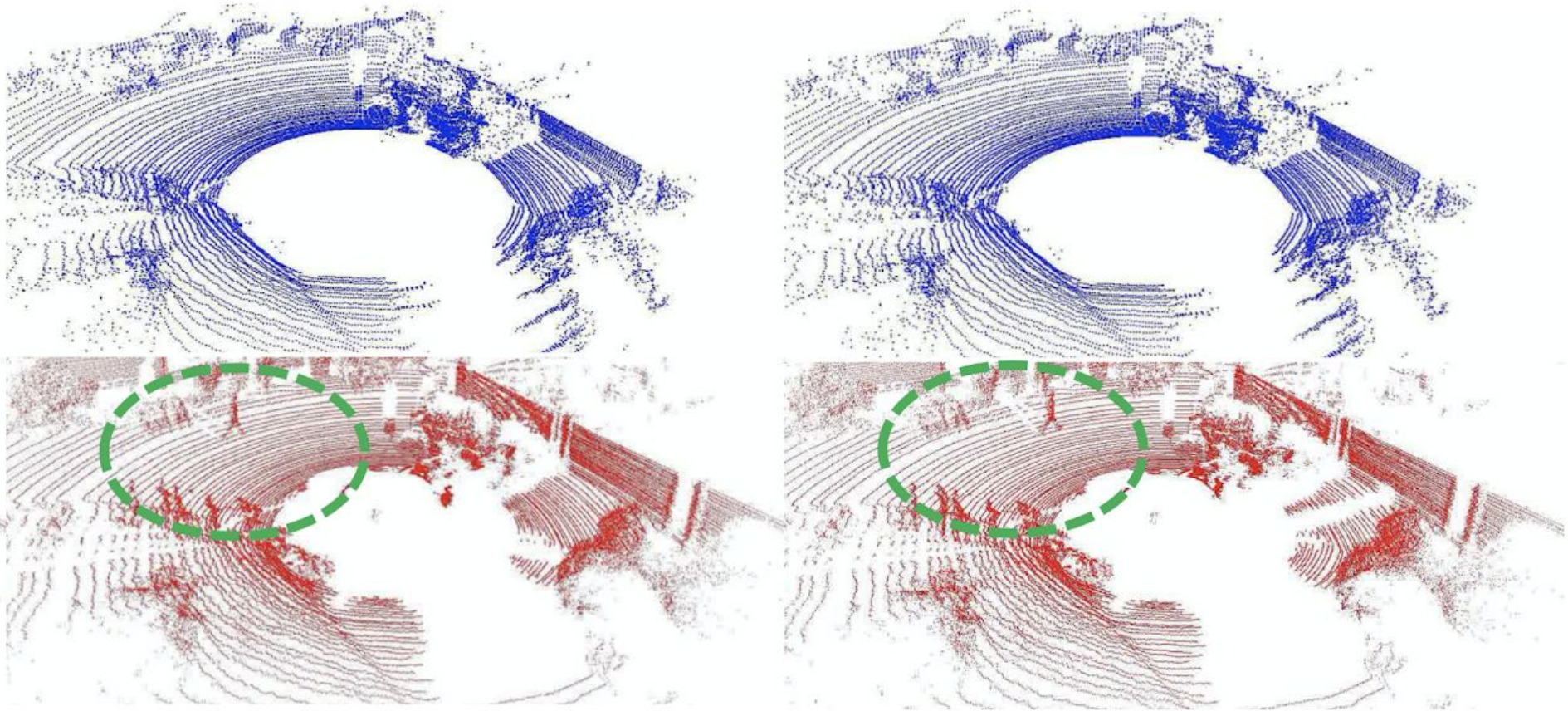
- More (unlabeled) data improves sensor forecasting.

Qualitative Results & Failure Cases





Qualitative Results & Failure Cases



Strengths & Weaknesses

Strengths

- Promising new approach to PnP.
- Acknowledges limitations of current prediction metrics.
- Reduced reliance on training data.
- Can improve many things about the architecture.
- Adding maps to the method is possible (we can predict egomotion).

Weaknesses

- GT track assignment may make method look better than it is.
- Fails badly on smaller objects like pedestrians (likely due to pooling).
- Still need supervision for the detector.
- Multi-modal predictions are much harder in this setting.

Predicted Questions

- So if this basically forecasting LiDAR flow?
  - Well, kind of, yes.
- What labels do they actually need?
  - Need supervised data to train the detector, then nothing.
- How fast is this?
  - They don’t say, but probably not too horrible when they use range images.
  - But need extra work to actually do prediction!

Insights

- Our PnP architecture would naturally bypass their global pooling bottlenecks.
- Detection does not need to be separate.
  - Can backprop through the forecasting to fine-tune a detector
- Residual forecasting may help
  - Decomposing frames into egomotion and dynamic object components can further constrain the task

Conclusion

- The paper defines a new task “sensor forecasting” and applies it to prediction
- Forecast sensor data, then treat prediction as tracking
- Highlights importance of new prediction metrics
- Decent forecasting without track labels
- Some limitations in the evaluation (baselines, GT matching)
- Lots of room for architectural improvement

Thank you!

Q&A Time!

Paper: https://arxiv.org/abs/2003.08376

Special thanks to Sergio and Julieta for proofreading help!

Bonus Slides

Architecture Details

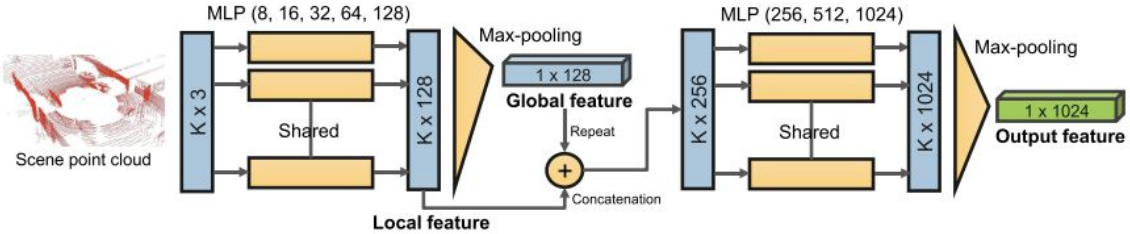


Fig. 4: Point-based encoder. A scene point cloud with shape of $K \times 3$ is fed into shared MLP to obtain local feature for each point. Then a global feature is obtained by max-pooling. We then fuse the local and global features by concatenation, which is then processed by subsequent MLP and max-pooling to obtain the output feature.

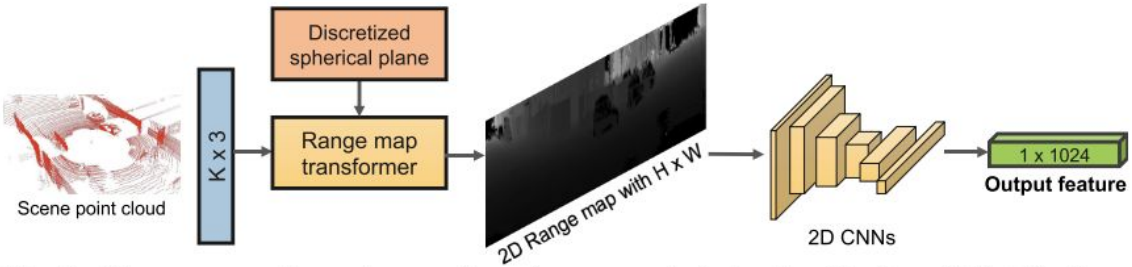


Fig. 5: Range map-based encoder. A scene point cloud with size of $K \times 3$ along with a discretized spherical plane is fed into the range map transformer to obtain the 2D range map with resolution of $H \times W$. We use a standard 2D CNNs to extract the final output feature from the 2D range map.

Results: Earth Mover Distance

- Proposed for point clouds in [A Point Set Generation Network for 3D Object Reconstruction from a Single Image]

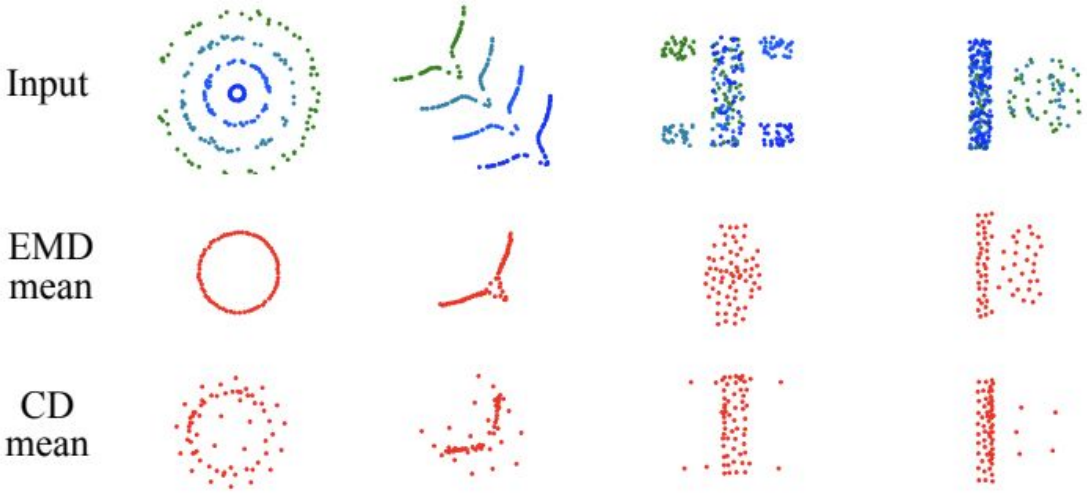
Earth Mover's distance Consider $S_1, S_2 \subseteq \mathbb{R}^3$ of equal size $s = |S_1| = |S_2|$. The EMD between A and B is defined as:

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

where $\phi : S_1 \to S_2$ is a bijection.

Approximated for computational reasons following the above paper.

Earth Mover Distance vs. Chamfer



In Figure 3, we illustrate the distinct mean-shape behavior of EMD and CD on synthetic shape distributions, by minimizing $\mathrm{E}_{s\sim\mathbb{S}}[L(x,s)]$ through stochastic gradient descent, where $\mathbb{S}$ is a given shape distribution, L is one of the distance functions.

In the first and the second case, there is a single continuously changing hidden variable, namely the radius of the circle in (a) and the location of the arc in (b). EMD roughly captures the shape corresponding to the mean value of the hidden variable. In contrast CD induces a splashy shape that blurs the shape's geometric structure. In the latter two cases, there are categorical hidden variables: which corner the square is located at (c) and whether there is a circle besides the bar (d). To address the uncertain presence of the varying part, the minimizer of CD distributes some points outside the main body at the correct locations; while the minimizer of EMD is considerably distorted.

Source: https://arxiv.org/pdf/1612.00603.pdf

Ablation Study: Full vs. Partial Sweeps

Table 3: Effect of the global scene constraint.

Datasets	Metrics	w/o Scene	w/ Scene
KITTI-1.0s	CD↓	3.37	0.89
	EMD↓	261.95	128.81
KITTI-3.0s	CD↓	5.91	0.94
	EMD↓	358.16	175.54
KITTI-1.0s	AADE↓	0.845	0.317
	AFDE↓	1.593	0.405
KITTI-3.0s	AADE↓	1.347	0.408
	AFDE↓	2.984	0.504

- Predicting full sweeps, not just points inside GT boxes helps.