Online Map Vectorization for Autonomous Driving: A Rasterization Perspective

Gongjie Zhang†[1] **Jiahao Lin**†[1] **Shuang Wu**†[1] **Yilin Song**[1] **Zhipeng Luo**[1,2]
Yang Xue[1] **Shijian Lu**[2] **Zuoguan Wang** ✉ [1]

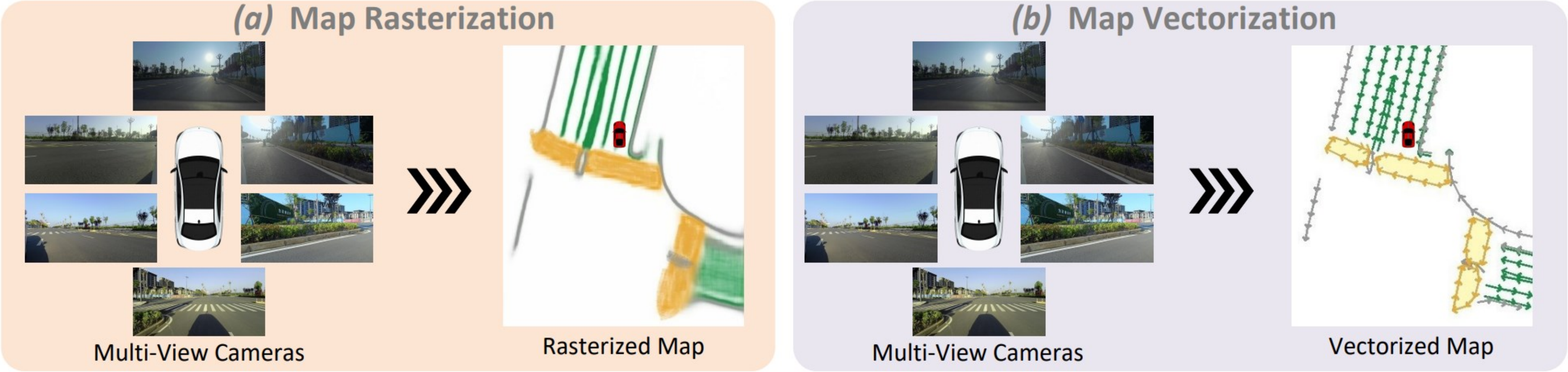
[1]Black Sesame Technologies [2]Nanyang Technological University, Singapore

† : equal contribution. ✉ : corresponding author.

Project Page: https://github.com/ZhangGongjie/MapVR

Background

- Online HD map construction is essential to autonomous driving.
- There are two types of online HD map construction:



- Generally, vectorized maps offer parameterized representations of road environment, thus can be better utilized by downstream planning & control tasks.

Background

- Map vectorization is an emerging topic in autonomous driving.
- Two primary challenges are noted:

1. How to **<u>precisely</u>** evaluate the performance of map vectorization?
2. How to learn **<u>precise</u>** map vectorization that meets the stringent requirements of autonomous driving?

Background

- How prior arts [1,2] address the two challenges:

1. How to **precisely** evaluate the performance of map vectorization?

→ Chamfer distance between two equidistant point sets.

2. How to learn **precise** map vectorization that meets the stringent requirements of autonomous driving?

→ To represent predictions and ground truths as equidistant points, then perform L1 loss supervision between them.

References:
[1] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. "VectorMapNet: End-to-end vectorized HD map learning." In ICML. 2023.
[2] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. "MapTR: Structured modeling and learning for online vectorized HD map construction." In ICLR. 2023

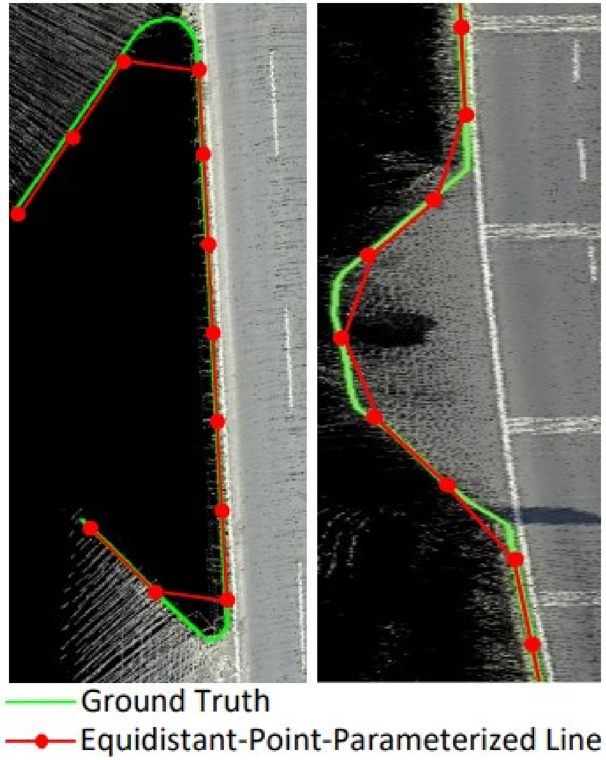
Background

- Equidistant points fail to meet the high-precision requirement of autonomous driving, in terms of both learning and evaluation.

(1) Equidistant points often cause parameterization error, especially at sharp bends or complex details.

(2) L1 loss and Chamfer distance on equidistant points tend to ignore fine-grained geometric details.

(3) Training with L1 loss on equidistant loss causes ambiguous supervision, as the intermediate points often lack clear visual clues.



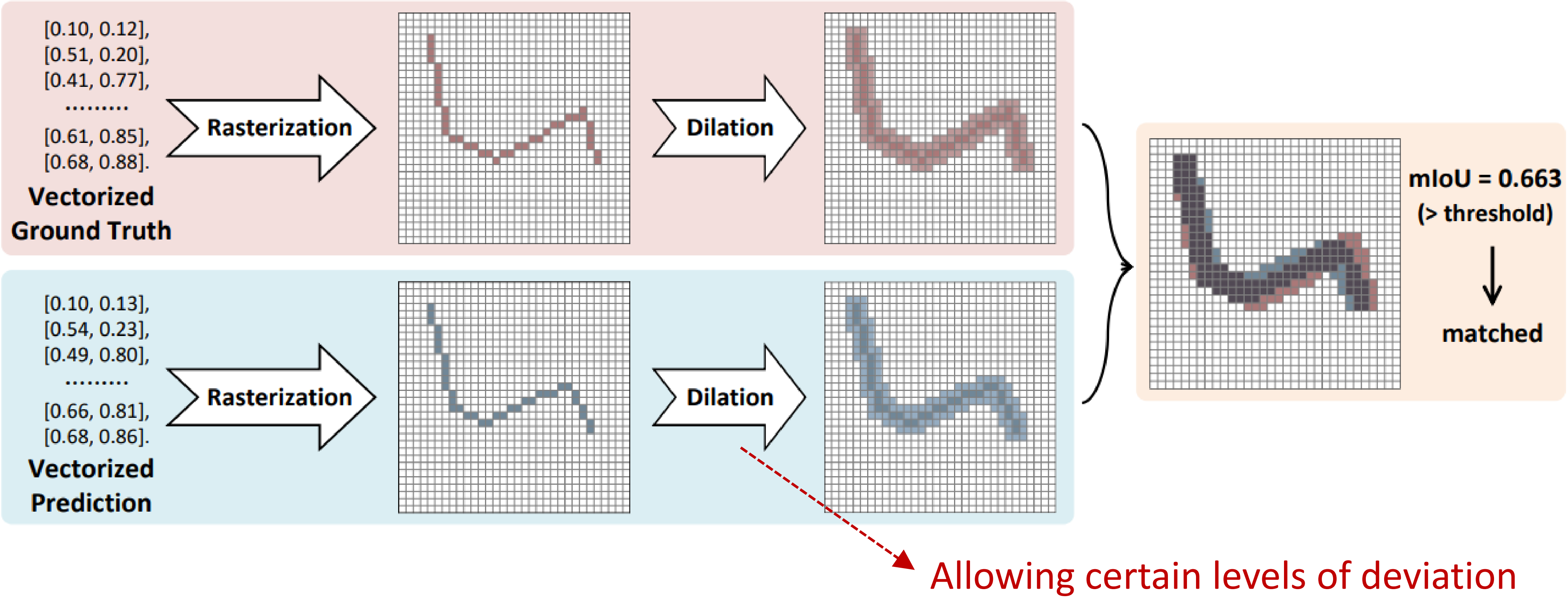
Our Contributions

- We re-introduce a rasterization perspective into the topic of map vectorization.

(1) **A rasterization-based map vectorization evaluation metric**, which exhibits increased sensitivity to minor deviations, providing more accurate and reasonable assessment of map vectorization performance.

(2) **MapVR (Map Vectorization via Rasterization)**, which enables precise and geometric-aware supervision for map vectorization through differentiable rasterization.

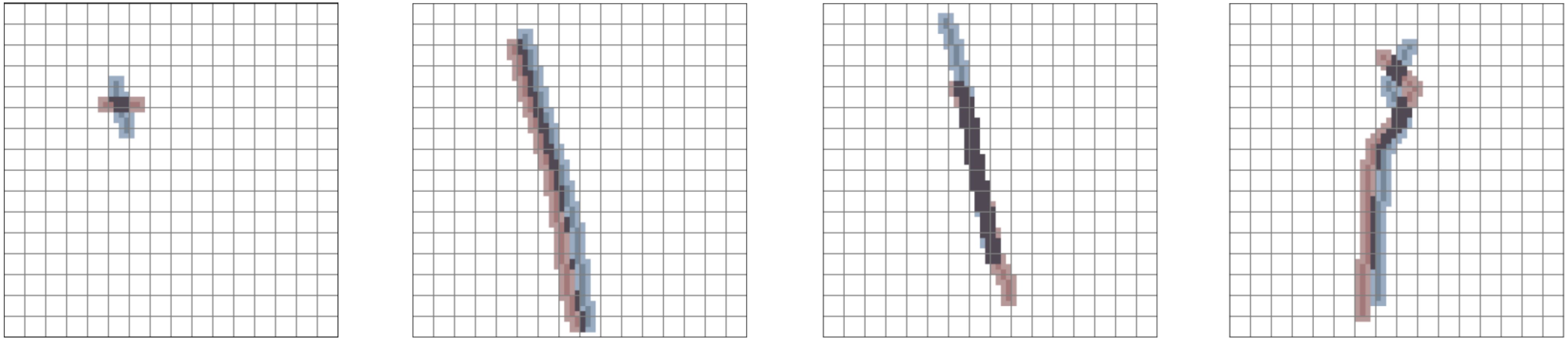
Rasterization-Based Evaluation Metric



Similar to MS-COCO's metric for instance segmentation, AP is calculated.

Rasterization-Based Evaluation Metric

- Our rasterization-based metric has better sensitivity to minor deviations and provides more reasonable evaluation compared with Chamfer-Distance-based metric.



(a) $D_{Chamfer} = 0.35$ → matched
mIoU = 0.18 → not matched

(b) $D_{Chamfer} = 0.35$ → matched
mIoU = 0.17 → not matched

(c) $D_{Chamfer} = 1.12$ → not matched
mIoU = 0.56 → matched

(d) $D_{Chamfer} = 0.75$ → matched
mIoU = 0.20 → not matched

Better sensitivity for short lanes

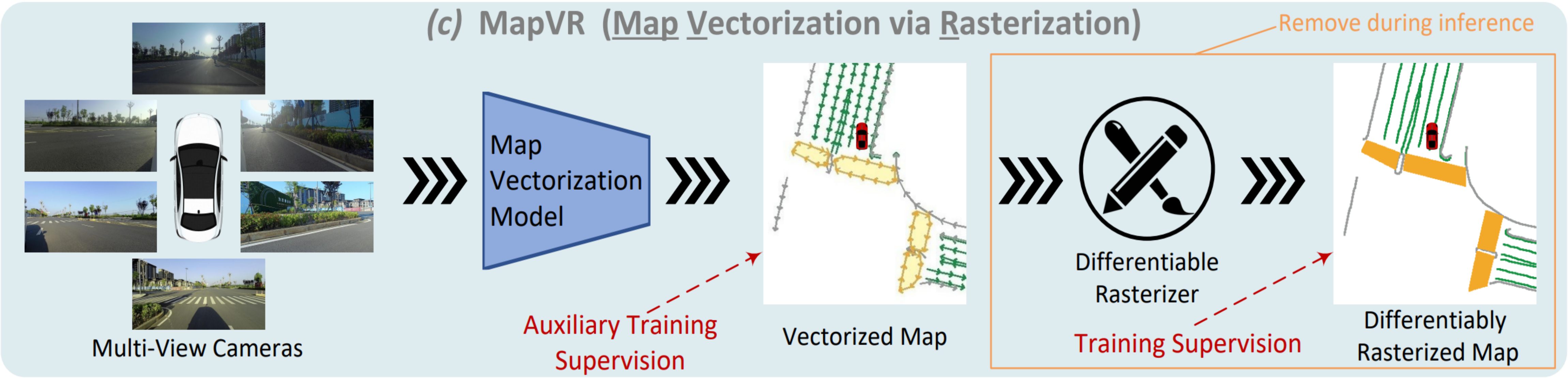
Better sensitivity for horizontal deviation

Robust against vertical deviation

Better sensitivity for minor deviation

MapVR (Map Vectorization via Rasterization)

Motivation: To combine the best of two worlds: *(i)* the simplicity of vectorized representation; *(ii)* the fine-grained training supervision of map rasterization.



MapVR (Map Vectorization via Rasterization)

$$I_{\text{line}}(x, y; P) = \exp\left(\frac{-D(x, y; P)}{\tau}\right)$$

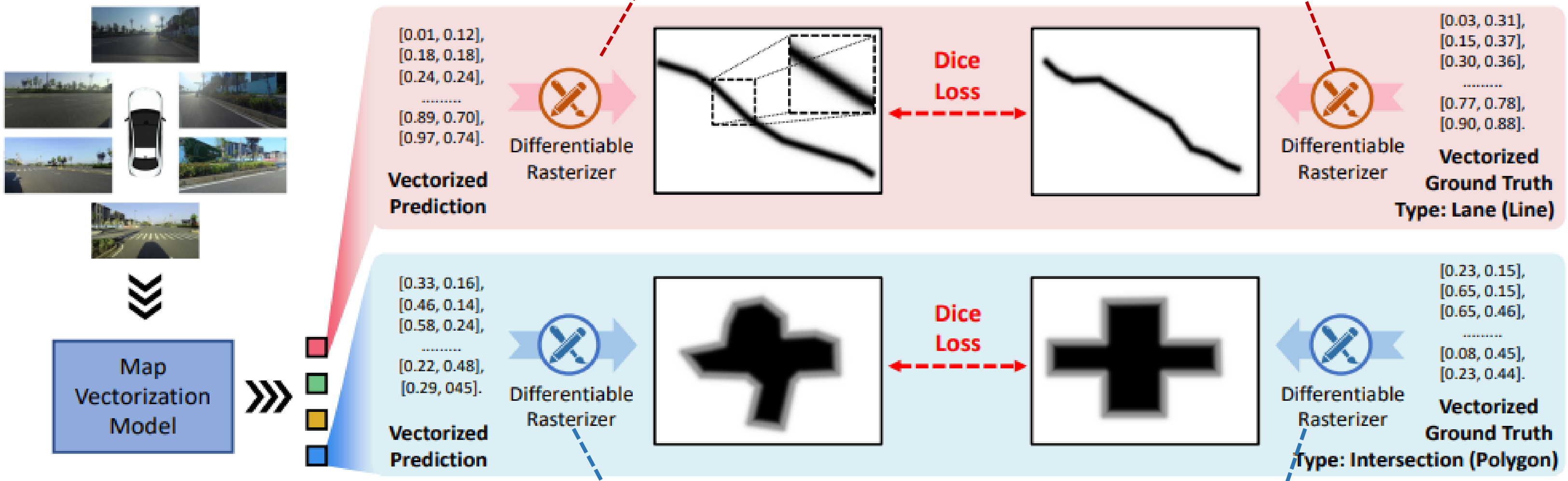


Figure : The learning pipeline of MapVR. MapVR utilizes a base model for vectorized map generation, followed by a customized differentiable rasterizer to produce HD maps, on which fine-grained, geometry-aware supervision is applied to enhance the precision of vectorized elements.

$$I_{\text{polygon}}(x, y; P) = \sigma\left(\frac{C(x, y; P) \cdot D(x, y; P)}{\tau}\right)$$

MapVR (Map Vectorization via Rasterization)

Additional Directional Regularization Loss

$$\mathcal{L}_{\mathrm{dir}} = \sum_{i=1}^{N-2} \frac{< \overrightarrow{P_i P_{i+1}}, \overrightarrow{P_{i+1} P_{i+2}} >}{|\overrightarrow{P_i P_{i+1}}| \cdot |\overrightarrow{P_{i+1} P_{i+2}}|}$$

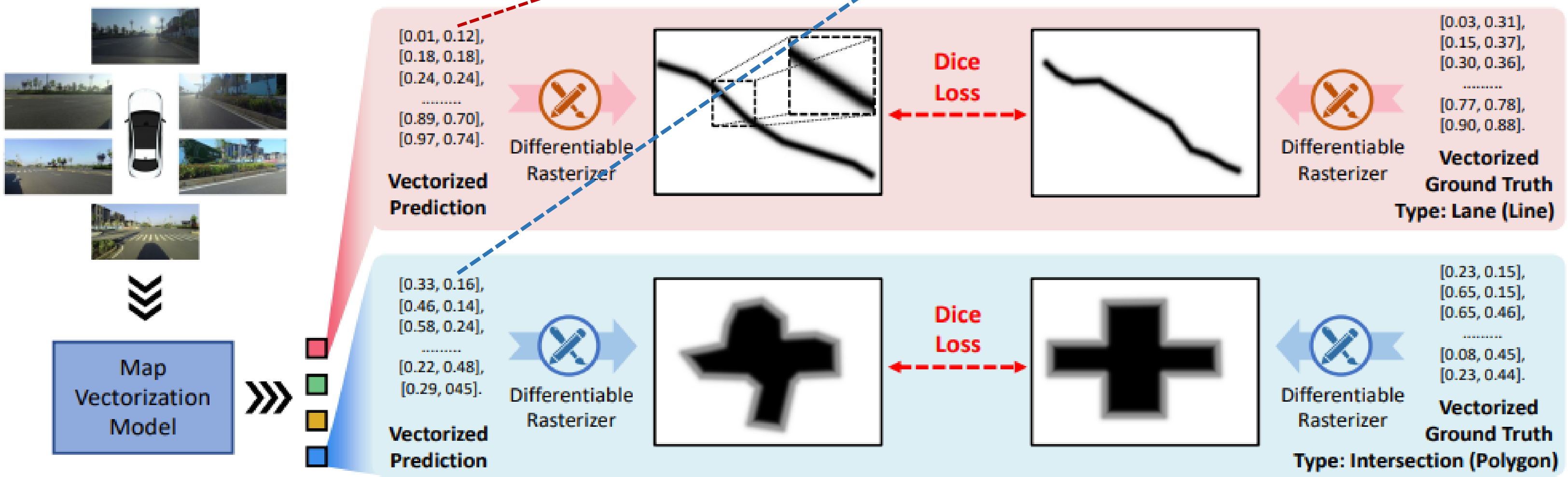
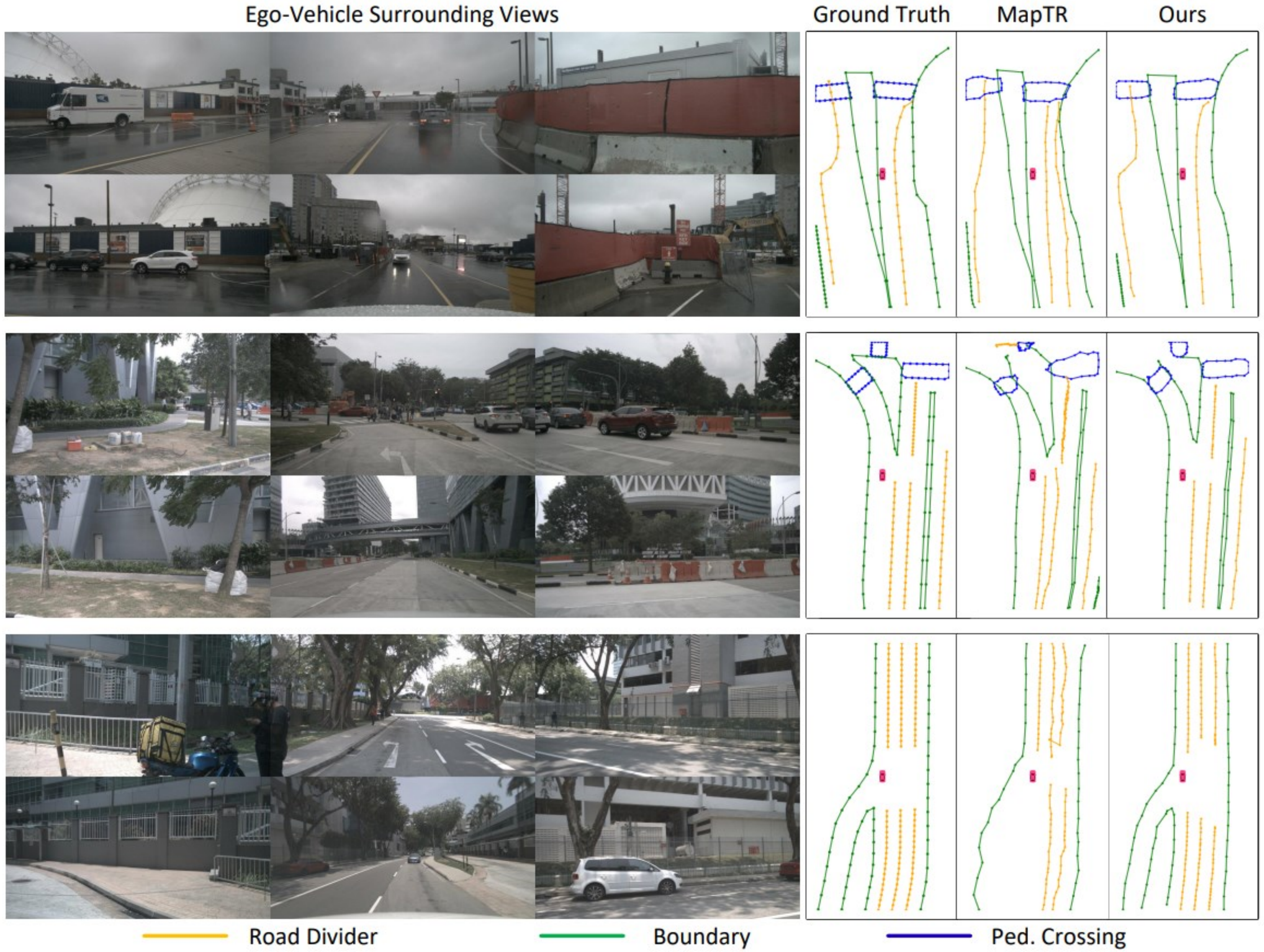


Figure : The learning pipeline of MapVR. MapVR utilizes a base model for vectorized map generation, followed by a customized differentiable rasterizer to produce HD maps, on which fine-grained, geometry-aware supervision is applied to enhance the precision of vectorized elements.

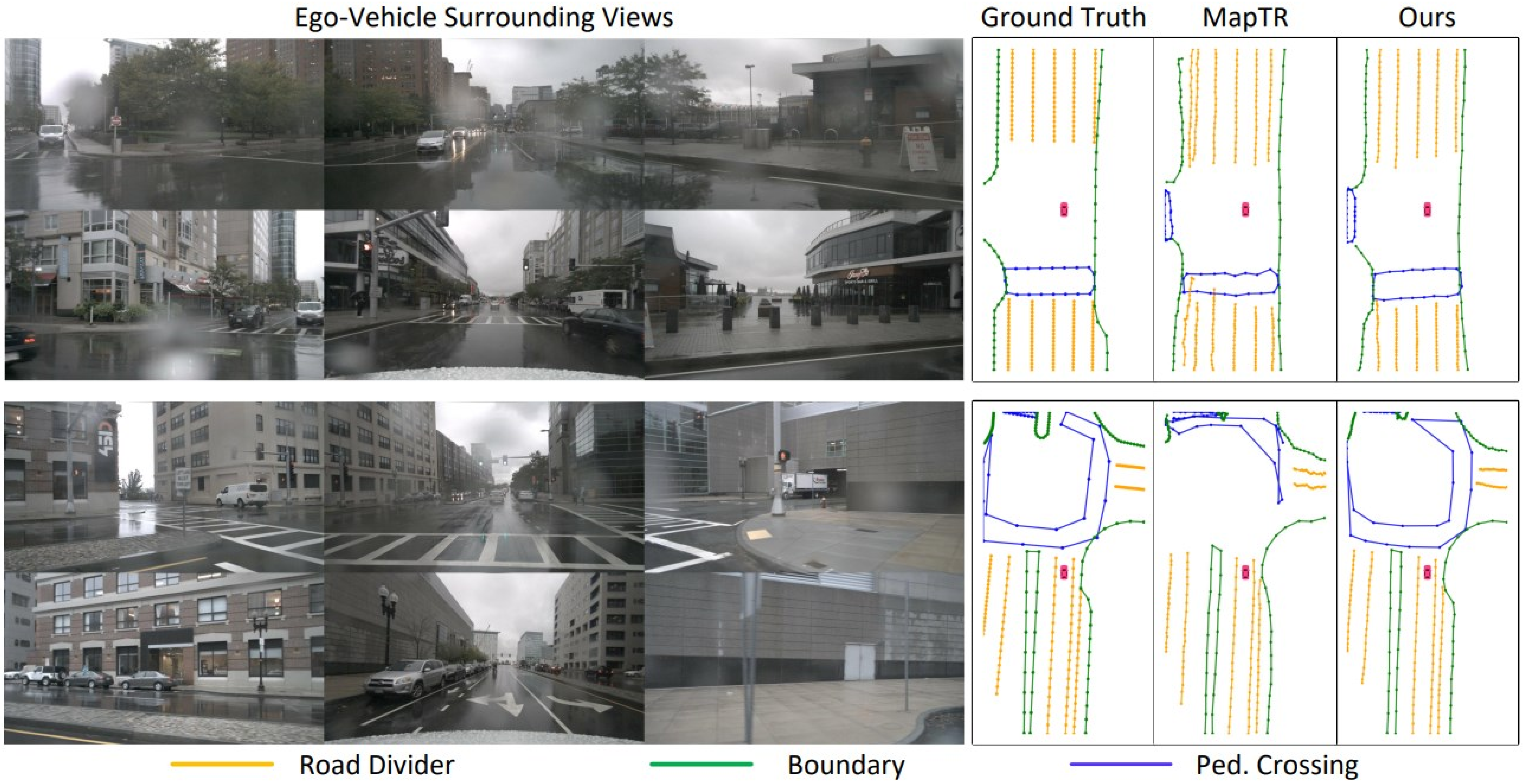
MapVR (Map Vectorization via Rasterization)

MapVR greatly improves the precision of map vectorization.



MapVR (Map Vectorization via Rasterization)

MapVR greatly improves the precision of map vectorization.



MapVR (Map Vectorization via Rasterization)

MapVR greatly improves the precision of map vectorization.

Table 1: Comparison of various map vectorization methods on nuScenes Map (basic) validation set.

Method	Modality	Backbone	#Epochs	$AP_{Chamfer}$				AP_{raster}				FPS
				ped	div	bdry	*avg.*	ped	div	bdry	*avg.*	
HDMapNet [16]	C	Effi-B0	30	14.4	21.7	33.0	23.0	-	-	-	-	0.8
HDMapNet [16]	C & L	Effi-B0	30	16.3	29.6	46.7	31.0	-	-	-	-	0.5
VectorMapNet [26]	C	Res-50	110	36.1	47.3	39.3	40.9	26.2	12.7	6.1	15.0	2.9
VectorMapNet [26]	C & L	Res-50	110	37.6	50.5	47.5	45.2	-	-	-	-	-
MapTR [20]	C	Res-50	24	46.3	51.5	53.1	50.3	32.4	23.5	17.1	24.3	**18.4**
MapTR [20]	C	Res-50	110	56.2	59.8	60.1	58.7	43.6	35.6	25.8	35.0	**18.4**
MapTR [20]	C & L	Res-50	24	56.4	61.8	**70.1**	62.7	46.4	39.2	50.0	45.2	7.2
MapTR [20] + MapVR (Ours)	C	Res-50	24	47.7	54.4	51.4	51.2	37.5	33.1	23.0	31.2	**18.4**
MapTR [20] + MapVR (Ours)	C	Res-50	110	55.0	61.8	59.4	58.8	46.0	39.7	29.9	38.5	**18.4**
MapTR [20] + MapVR (Ours)	C & L	Res-50	24	**60.4**	**62.7**	67.2	**63.5**	**52.4**	**46.4**	**54.4**	**51.1**	7.2

- *In modality, 'C' denotes multi-view camera input and 'C & L' denotes combined multi-view camera and LiDAR input.*
- *When LiDAR data is incorporated, PointPillars [14] serves as the backbone for processing the LiDAR data.*
- *The abbreviations 'ped', 'div', and 'bdry' correspond to the map elements of pedestrian crossing, divider, and boundary, respectively.*

Table 2: Map vectorization performance on nuScenes Map (extended) validation set.

Method	$AP_{Chamfer}$							AP_{raster}							FPS
	ped	stp	int	cap	div	bdry	*avg.*	ped	stp	int	cap	div	bdry	*avg.*	
MapTR [20]	34.3	29.9	21.5	37.9	44.9	45.1	35.6	22.5	12.1	38.4	23.4	18.3	12.1	21.1	**18.4**
MapTR [20] + MapVR (Ours)	**39.5**	**31.6**	**21.9**	**42.4**	**45.8**	**45.9**	**37.9**	**30.8**	**13.9**	**43.3**	**32.8**	**27.0**	**18.8**	**27.8**	**18.4**

- *All competing methods take multi-view cameras as input, use ResNet-50 [11] as the backbone, and are trained for 24 epochs.*
- *'ped', 'stp', 'int', 'cap', 'div', and 'bdry' denote pedestrian crossing, stopline, intersection, carpark area, divider, and boundary, respectively.*

MapVR (Map Vectorization via Rasterization)

MapVR greatly improves the precision of map vectorization.

Table 3: Comparison of various map vectorization methods on Argoverse2 validation set.

Method	$AP_{Chamfer}$				AP_{raster}			
	ped	div	bdry	*avg.*	ped	div	bdry	*avg.*
HDMapNet [16]	13.1	5.7	37.6	18.8	-	-	-	-
VectorMapNet [26]	38.3	36.1	39.2	37.9	-	-	-	-
MapTR [20]	**54.7**	**58.1**	56.7	56.5	22.1	32.6	24.0	26.2
MapTR [20] + MapVR (Ours)	54.6	**60.0**	**58.0**	**57.5**	**23.5**	**36.5**	**30.2**	**30.1**

•*'ped', 'div', and 'bdry' denote pedestrian crossing, divider, and boundary, respectively.*

Table 4: Map vectorization performance on 6V-mini-v0.4 dataset (our proprietary commercial dataset).

Method	AP_{raster}				
	lane	curbside	stopline	crosswalk	intersection
MapTR [20]	41.3	32.9	7.6	13.3	43.6
MapTR [20] + MapVR (Ours)	**50.8**	**37.3**	**11.8**	**14.3**	**44.0**

MapVR (Map Vectorization via Rasterization)

Ablation Study

Table 5: MapVR's ablation experiments on nuScenes Map (basic) validation set. All models employ ResNet-50 as backbones and are trained for 24 epochs. MapVR's default setups are marked in gray.

(a) Rasterization resolution. '×' denotes no rasterization.

resolution	×	64x32	128x64	180x90	256x128	320x160
mAP_{raster}	24.3	21.5	29.8	30.4	**31.2**	30.9
$mAP_{Chamfer}$	50.3	45.1	/	50.6	**51.2**	50.9

(b) Line rasterization softness τ.

line softness τ	0.5	1.0	2.0	4.0	6.0
$mAP_{raster(divider)}$	29.2	31.7	**33.1**	32.5	31.4
$mAP_{Chamfer(divider)}$	48.0	50.3	**54.4**	53.3	52.8

(c) Regularization on point direction.

regularization	None	w/ GT	w/ self
mAP_{raster}	29.5	29.3	**31.2**
$mAP_{Chamfer}$	48.5	48.5	**51.2**

(d) Rasterization geometry-awareness.

	all as lines	lines & polygons
$mAP_{raster(ped\ xing)}$	21.8	**37.5**
$mAP_{Chamfer(ped\ xing)}$	34.9	**47.7**

(e) MapVR *vs.* parallel segm.

	MapVR	parallel segm
mAP_{raster}	**31.2**	26.7
$mAP_{Chamfer}$	**51.2**	48.1

- *$mAP_{Chamfer}$ are added upon reviewers' kind suggestions. Entries marked with '/' are unavailable due to accidentally deleted checkpoints.*

MapVR (Map Vectorization via Rasterization)

MapVR only adds slight extra training computational cost.

MapVR adds no inference cost.

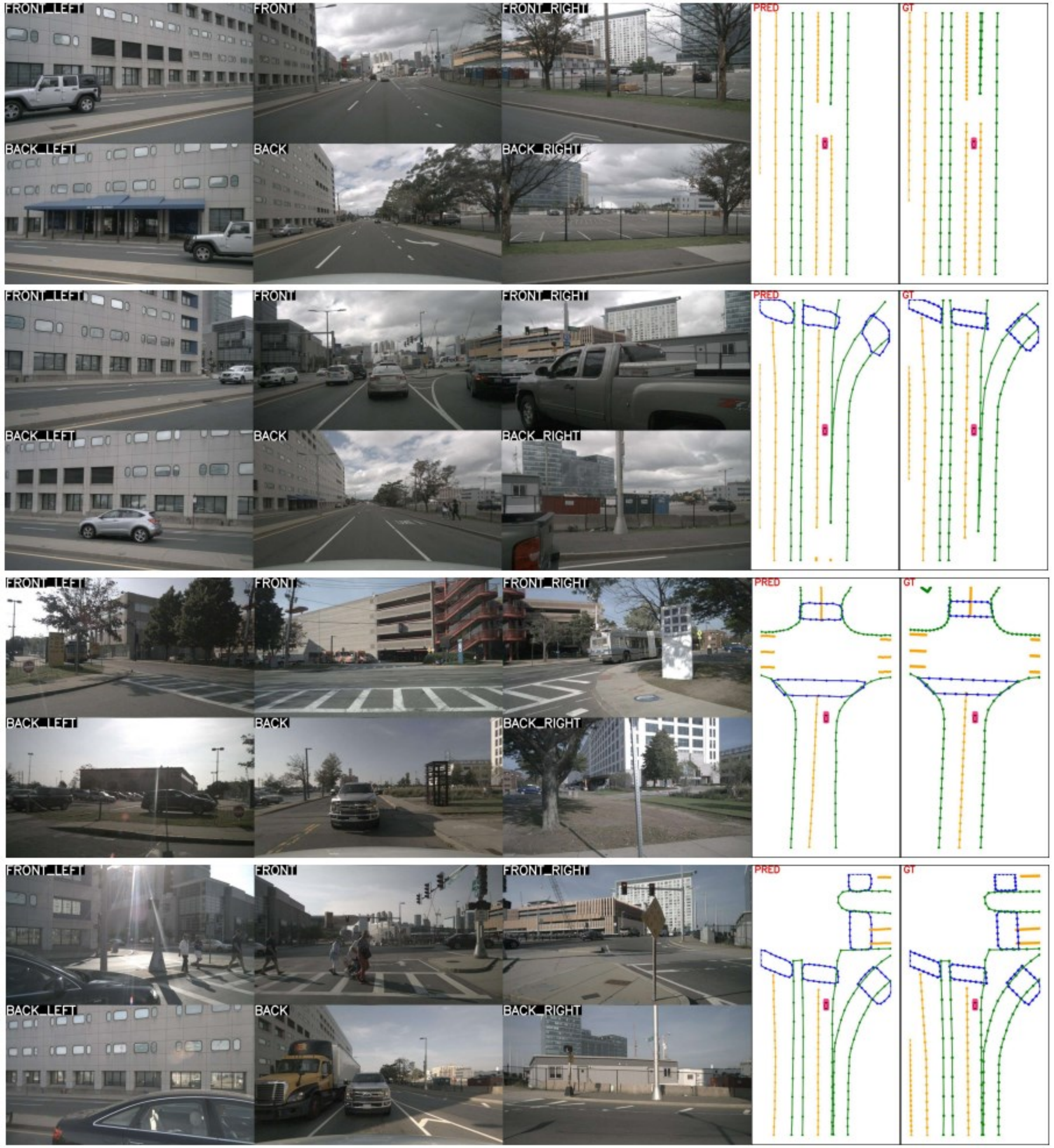
Table 6: MapVR's computational overhead during the training stage. Results were obtained with 8x NVIDIA A100 GPUs under the same training setups.

Method	Modality	Backbone	Training Time / Iter	GPU Memory Usage
MapTR [20]	C	Res-50	0.82 s	14021 MB
MapTR [20]	C & L	Res-50	1.18 s	28557 MB
MapTR [20] + MapVR (Ours)	C	Res-50	0.91 s	14169 MB
MapTR [20] + MapVR (Ours)	C & L	Res-50	1.37 s	28673 MB

- *In modality, 'C' denotes multi-view camera input and 'C & L' denotes combined multi-view camera and LiDAR input.*

Summary

- Vectorized representation can benefit from a rasterization perspective in terms of evaluation and learning.
- We hope our work can serve as basis for map vectorization that suits the stringent requirement of autonomous driving!



Thank you!

We are committed to ensure the reproducibility of our algorithms through open-sourcing.

SCAN ME

Codes for MapVR

SCAN ME

Codes for Standalone Evaluation Toolkit