Localization Is All You Evaluate: Data Leakage in Online Mapping Datasets and How to Fix It

Adam Lilja[1,2] Junsheng Fu[2] Erik Stenborg[2] Lars Hammarstrand[1]
[1]Chalmers University of Technology [2]Zenseact
{firstname.lastname}@{chalmers.se, zenseact.com}

Abstract

The task of online mapping is to predict a local map using current sensor observations, e.g. from lidar and camera, without relying on a pre-built map. State-of-the-art methods are based on supervised learning and are trained predominantly using two datasets: nuScenes and Argoverse 2. However, these datasets revisit the same geographic locations across training, validation, and test sets. Specifically, over 80% of nuScenes and 40% of Argoverse 2 validation and test samples are less than 5 m from a training sample. At test time, the methods are thus evaluated more on how well they localize within a memorized implicit map built from the training data than on extrapolating to unseen locations. Naturally, this data leakage causes inflated performance numbers and we propose geographically disjoint data splits to reveal the true performance in unseen environments. Experimental results show that methods perform considerably worse, some dropping more than 45 mAP, when trained and evaluated on proper data splits. Additionally, a reassessment of prior design choices reveals diverging conclusions from those based on the original split. Notably, the impact of lifting methods and the support from auxiliary tasks (e.g., depth supervision) on performance appears less substantial or follows a different trajectory than previously perceived. https://github.com/LiljaAdam/geographical-splits

1. Introduction

A core capability for an autonomous vehicle is to estimate the road in its vicinity. There are two complementary approaches for this task: retrieving the information from a pre-built map using localization [5], and directly predicting the online map using onboard sensors like camera and lidar [19]. The former, *Online Map Retrieval* (OMR), assumes there exists a map over the deployment area, while the latter, *Online Map Estimation* (OME) assumes no such map exists. A pre-built map provides detailed information but also requires robust localization and continuous map updates to be useful. OME sidesteps this and instead relies solely on onboard sensors and algorithms. It is thus independent of variations in current surroundings compared to mapped data. The challenge with OME instead lies in generalizing to new locations, beyond the places captured in the training data.

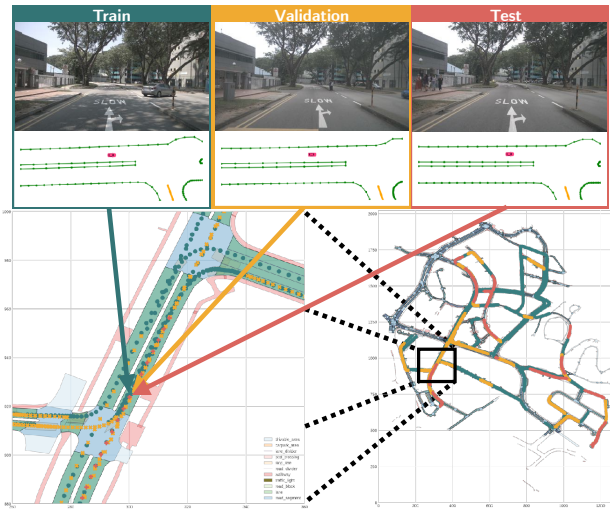


Figure 1. Example of substantial geographical overlap between train, val, and test sets for in nuScenes'. Green circle, Orange cross, and Red plus are training, validation, and test samples.

The current state-of-the-art methods for online mapping are based on supervised learning. While there exist large public datasets [1, 2, 6, 12, 33, 36, 38] that support training of perception and planning models for many of the crucial tasks of an autonomous vehicle, only a few of these provide the HD maps needed to train online mapping models, see Tab. 1. Moreover, as these datasets are mainly constructed to support object detection, object tracking, and motion forecasting tasks, we argue that they, in their original form, are not ideal for training online mapping models for two reasons: (1) The training, validation, and test sets are constructed by splitting the data temporally. This is an easy way to ensure that, *e.g.*, the same vehicle is not present in the same position in the training and test sets yielding fair evaluation of object detection methods. However, since the

Figure 2. Example from nuScenes where input images, predictions, and ground truth for a test sample are displayed together with the ground truth of the closest train sample. Lane dividers, road boundaries, and pedestrian crossings are visualized in orange, green, and blue.

areas where these datasets are collected are relatively small, the same areas are revisited multiple times. Failing to account for this when dividing the data, results in significant geographical *inter-set* overlap between the training, validation and test sets as Fig. 1 exemplifies. Fig. 2 further visualizes the input images, prediction, ground truth, and the closest training sample of a test sample for another example. The test sample is very close to the nearest training sample, enabling a method to score well on that test sample by memorizing the image-to-map connection from the training sample, and recalling it at test time. This connection can be created using any features in the images such as buildings and traffic signs. This would rather resemble localizing in a pre-built map and presenting it at test time (OMR), than the intended task of online mapping (OME). (2) As each data sample is collected from a data sequence under normal driving, there is substantial *intra-set* overlap/correlation between the data samples within the sets. This correlation is especially evident near intersections where the vehicle is standing still or driving slowly. Essentially, the ground truth HD maps for close training samples are only slight transformations of the same information. Both these aspects violate the independent and identically distributed data assumption fundamental in training machine learning models.

In this paper, we show that there is a significant geographical overlap in two of the most commonly used datasets for online mapping, nuScenes [2] and Argoverse 2 [36]. Furthermore, we show that this overlap causes severe inflation in the reported performance of the state-of-the-art methods and, more critically, results in poorer generalizability than initially perceived.

To support future research in OME we provide geographically disjoint splits for both nuScenes and Argoverse 2. We provide splits under two slightly different problem settings, *Near Extrapolation* where we assume we have training data from the same neighbourhood/city and *Far Extrapolation* where training and test data are from separate cities. The former is an easier setting and can be viewed as a proper substitution for the original splits while the latter enables the exploration of the more relevant question; how well do these methods generalize to new environments with larger distribution shifts?

We re-evaluate state-of-the-art methods trained on these splits to give a more representative view of their performance. Additionally, we perform more detailed experiments to investigate how the large overlap has affected conclusions regarding important algorithmic choices.

2. Related Work

This section introduces the online mapping setting (OME), highlights key methodologies and datasets used for training and evaluation. While the OME field is relatively recent, the utilization of machine learning in processing geospatial data (GeoML) has a longer history. Hence, we also provide a brief overview of GeoML, noting parallels between its challenges and those encountered in OME.

2.1. Online Mapping

The current online mapping methods are either segmentation-based [7, 9, 14, 16, 17, 21, 23, 24, 27, 37, 40] or vector-based [16, 18–20, 30]. The main difference lies in how the online map is represented. In segmentation-based maps, the aim is to predict a rasterized grid where each cell is classified as, e.g., empty, lane marking or road edge. For vector-based methods, the predicted map is described by a set of objects with a given class and the geometry is described by a vector of point coordinates, *i.e.*, a polyline.

Both categories universally adopt a core technique known as *lifting*. This entails converting image features from perspective view (PV) to Bird's Eye View (BEV) features, from which the map is predicted. The primary challenge for these methods lies in accurately mapping features from the perspective view to their corresponding locations in BEV due to the absence of depth information in images. Various lifting methods have been proposed, broadly categorized as either *pulling* the features to BEV from PV, or *pushing* the features from PV to BEV.

In essence, *pulling* methods retrieve features from PV

based on dense queries in BEV [7, 17, 23, 41]. A straightforward approach is the Inverse Perspective Mapping (IPM) [22] and involves projecting predefined points in BEV into PV using camera parameters and interpolating features from these projected positions. Alternatively, methods like Geometry-Guided Kernel Transformer (GKT) [7] and BEVFormer [17] use a combination of geometry and attention mechanisms to pull features to BEV-space efficiently. In contrast, Cross-View Transformer (CVT) [41] pulls features without an intermediate BEV representation using cross-attention with a canonical form of all camera views.

The *pushing* methods used in, *e.g.* [9, 16, 21, 24, 27, 37], specialize in learning how to map PV features to BEV. Among them, depth-based approaches aim to learn the depth distribution for each image pixel to project PV features accurately. For instance, LSS [24] tries to learn a categorical depth distribution for each pixel and use it to weigh how much the corresponding PV feature should influence the corresponding BEV-cell. Pyramid Occupancy Network (PON) [27] uses a multi-scale dense transformer for low-resolution BEV projection, employing deconvolutions for upsampling predictions, whereas HDMapNet [16] learns BEV projection through a Multilayer Perceptron (MLP).

These lifting methods have been successfully adapted to a segmentation head for online mapping. Also the vector-based approach, introduced in HDMapNet [16] and further developed in works such as [18–20, 30], have shown great promise by utilizing network heads inspired by the object-detection community. While HDMapNet uses a handcrafted post-processing step, subsequent methods are instead end-to-end trainable. For example, VectorMapNet [20] uses IPM [22] for lifting image features to BEV from which a transformer decoder predicts coarse object representations. These are then refined in a joint Autoregressive Transformer (ART) that attends the coarse prediction and all BEV features. MapTR [18] utilizes GKT [7] for lifting and a DETR-like [4] transformer decoder for predicting the objects. They use deformable attention to attend BEV features with hierarchical queries to predict a collection of objects defined by a set of points. MapTRv2 [19] builds on its predecessor, but uses LSS[24] for lifting and adds PV depth estimation and segmentation in both PV and BEV as auxiliary supervision. Lastly, StreamMapNet [39] uses BEVFormer-lifting, multi-point attention, and temporal information fusion.

All these methods, except [27, 39], are primarily evaluated on the original nuScenes and Argoverse 2 splits with considerable inter-set overlap. The validity of conclusions drawn regarding their performance on online mapping tasks is thus severely limited. To give a fairer view of their performance on the intended problem setting, we re-evaluate them on our proposed splits and analyze the results.

Dataset	Split	Source	Main Map Purpose	#Samples			Geo. Split
				Train	Val	Test	
nuScenes [2]	Original	nuSc	OD/MF	28k	6k	6k	✗
Argoverse 1 [6]	Original	argo1	OD/MF	39k	15k	13k	✓
Argoverse 2 [36]	Original	argo2	OD/MF	110k	24k	24k	✗
Waymo [33]	Original	way	OD/MF	122k	30k	40k	✗
nuScenes [2]	**Near**	nuSc	OM	28k	6k	6k	✓
Argoverse 2 [36]	**Near**	argo2	OM	110k	24k	24k	✓
nuScenes [2]	**Far-A**	nuSc	OM	30k	9k	-	✓
nuScenes [2]	**Far-B**	nuSc	OM	31k	9k	-	✓
Argoverse 2 [36]	**Far-A**	argo2	OM	110k	46k	-	✓
Argoverse 2 [36]	**Far-B**	argo2	OM	101k	55k	-	✓
Argoverse 2 [36]	**Far-C**	argo2	OM	101k	55k	-	✓

Table 1. Datasets used for online mapping. The proposed splits are shown in bold. OD = object detection, MF = motion forecasting, OM = online mapping.

Split	nuScenes		Argoverse 2	
	Val	Test	Val	Test
Orig.	79.4%	85.5%	45.0%	41.9%
Near	0.9%	1.1%	0.0%	0.0%

Table 2. Ratios of validation and test samples within 5 m of training samples. The Near Extrapolation split has negligible overlap compared to the Original (Orig.) split for both datasets.

2.2. Online Mapping Datasets

A summary of datasets used for online mapping is provided in Tab. 1. Three original datasets provide the HD-maps required to enable the training of online mapping models, Argoverse 1 and 2 [6, 36], nuScenes [2] and Waymo [33]. All these primary datasets are mainly intended for object detection and motion forecasting tasks, and, in addition to supplying HD maps, these datasets provide rich annotations for dynamic objects. The predefined data splits provide fair and consistent evaluations across studies, but were originally designed for dynamic object perception rather than online mapping. They are temporally divided to prevent sample overlap across sets within a sequence, but do not ensure geographic separation. Despite this, nuScenes [2] and Argoverse 2 [36] are widely used for training online mapping models and have become the *de facto* standard. For example, online mapping methods using nuScenes include [3, 7, 9, 14–17, 21, 23–25, 27, 29, 37, 40, 41] and Argoverse 2 is used in [18–20, 30].

The nuScenes dataset contains 1 000 driving sequences collected in two cities (Boston and Singapore) with an area coverage of about 5 km^2 [33] and captures different types of city roads as well as containing diverse weather and illumination conditions. In total, the sequences consist of $40,000$ key-frame samples at a rate of 2 Hz, accompanied by object annotations. Additional sensor data is present between these key-frame samples, albeit without any object annotation. Online mapping methods typically adhere to the convention established by object detection methods, utilizing only key-frame samples for training. Furthermore, these

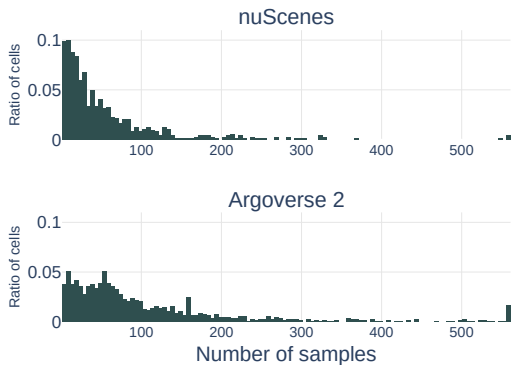


Figure 3. Number of samples within a cell with length 60 m. Argoverse 2 has higher *intra-set* sample density.

samples are closely together across the different sets as the same geographical position is re-visited multiple times. Fig. 1 illustrates a single example, where the proximity of validation and test samples to training samples is evident. Further analysis, as Tab. 2 depicts, reveals that approximately 80% and 85% of validation and test samples, respectively, are within 5 meters of a sample used during training.

The other prominently used dataset, Argoverse 2, is an extension of the 2019 version Argoverse [6]. In contrast to its predecessor, Argoverse 2 is not geographically split, but is much larger and collected from 6 U.S. cities with an area coverage of 17 km^2 [1]. It comprises $1\,000$ annotated driving sequences, which are on average 15 s long and annotated at 10 Hz. Each sample offers observations from similar sensors as nuScenes, albeit in a slightly different configuration, and the provided HD maps are in 3D, but with a focus on drivable area, road boundaries, lane dividers, and pedestrian crossings. By inspecting Argoverse 2, one can see that it also suffers from a considerable level of inter-set geographical overlap. Tab. 2 shows that approximately 45% and 42% of the validation and test samples are within a 5 m range of the closest train sample.

Comparing nuScenes and Argoverse 2, we note that the inter-set sample overlap is larger for nuScenes, and that Argoverse 2 boasts a larger number of samples and a more extensive coverage area. Another difference arises from Argoverse 2 being more densely sampled, resulting in less spatial variation and a higher intra-set sample density. The discrepancy is highlighted in Fig. 3, where samples are discretized into cells with a side length of 60 m, the typical evaluation range for most online mapping methods, *e.g.* [16, 18–20, 30]. The distribution of non-empty cells over the number of samples they contain has more probability mass with the higher counts for Argoverse 2. Despite Argoverse 2 having nearly four times the number of samples than nuScenes, the number of non-empty cells is only 1.7 times higher.

2.3. Machine Learning on Geospatial Data

Utilizing machine learning for geospatial applications has been a longstanding challenge. The choice of geographic regions for training and evaluation significantly impacts the evaluation outcomes and, as highlighted in [28], it is crucial to select regions that mirror the desired goals for an accurate assessment. Neglecting geographic considerations in evaluating these models can thus yield inadequate and inaccurate assessments of their performance. This issue, characterized by geographic overlap between training and evaluation data, has been observed in various domains, including satellite image modeling [11], ecological mapping [10], medical imaging [34], and 3D object detection [31].

In [26], the authors discuss how deviating from the assumption of independent training and evaluation data can lead to favouring complex models. They propose a solution known as block cross-validation, where data is strategically split geographically to mitigate these concerns. Similarly, [32] advocates for spatially partitioning the data instead of random allocation to bolster the independence between training and validation datasets used for cross-validation.

In the online mapping community, [27] acknowledged the need for a geographically divided train and test split when evaluating their method (PON) on nuScenes. They coarsely partition the sets along large, visually similar, neighbourhoods, limiting the diversity within each set. Furthermore, their proposed geographical split does not take the validation set into account and discards samples from the original test set reducing the total size of the dataset. Another approach is to divide training and evaluation according to the different cities as explored for nuScenes in [25]. Although being a valid split set, the distribution shift is large, and significantly increases the difficulty of the problem. Since only one data split is proposed, cross-validation cannot be performed. For Argoverse 2, [39] proposes a split that divides the training and validation samples according to geographical positions, but does not allocate a geographically separate testing set. A separate test set representing unseen data is crucial for unbiased evaluation; hyperparameters should not be optimized specifically for this set. Hence, the need for robust data partitioning persists.

3. Geographically Disjoint Splits

In the online mapping setting, the difficulty level varies depending on assumptions about the geographic distribution shift between training and target data, ranging from near to far extrapolation. Near extrapolation assesses performance within the same cities as the training data, while far extrapolation evaluates generalization to cities beyond those in the training set. We believe that addressing the latter scenario, which poses a greater difficulty, should be the primary long-term target of the OME field.

3.1. Near Extrapolation

We created balanced geographically disjoint sets for training, validation, and testing in nuScenes and Argoverse 2 by partitioning data based on sample locations, reducing inter-set overlap. We employ the original splits' testing data as the map is provided for those samples. Split proportions are 70%, 15%, and 15% for training, validation, and test sets. To preserve diversity in zone classes (in the urban planning sense, e.g., residential, commercial, and industrial) while maintaining the frequency of road object classes, weather conditions, and time of day, fine-grained and thorough partitioning is performed. Our splits ensure proportional representation of samples from various criteria in each set, mirroring the full dataset's distribution and ensuring representation from all cities. Regions were defined manually based on map attributes, and splits are visualized in Appendix B and available on the project webpage.

While partitioning the data, we did not account for the intra-set overlap (Sec. 2.2). We do, however, acknowledge its potential importance and believe it warrants consideration in the utilization of these datasets. The specifics of how to address this concern and the associated implications are left for future research. We realize that the use of original test data has implications for multi-task networks (e.g. also performing object detection) and have ensured that balance remains when removing the original test data. The number of samples in each set can be seen in Tab. 1.

nuScenes Tab. 2 displays the ratio of validation and test samples located closer than 5 m of a training sample for the Near Extrapolation split. Our suggested splits show only minimal overlap. The remaining overlap is due to the samples close to cut-off regions between sets. To see the effects of these samples we conduct, in Appendix A, experiments where these samples have been filtered out and note that their impact is negligible. Further, we show that the weather conditions and time of day are equally distributed through the sets in our Near Extrapolation splits.

Argoverse 2 As Tab. 2 presents, no validation or test samples lie within 5 m of a training sample for the proposed split. Attributes concerning the conditions associated with the different sequences are not available for Argoverse 2 making it hard to do a quantitative analysis. Our main focus is thus to give a balanced geographically separated split, partitioning areas with similar zone classes equally in the different sets. Appendix B illustrates the distribution of the number of samples in each city as well as highlights the diversity of geographical distribution.

3.2. Far Extrapolation

Far Extrapolation through city-wise data splitting introduces a greater distribution shift between training and evaluation. A subset of cities from each dataset is designated for training, while the remainder is allocated for evaluation. Notably, there is an uneven city-wise sample distribution in both datasets, with, for instance, Boston containing 55% of samples in nuScenes, and Miami and Pittsburgh each constituting 35% of the Argoverse 2 data. To mitigate this imbalance, cities are grouped to achieve approximately equal-sized folds, with the training set comprising 70% of the data. Refer to Tab. 3 for the proposed city-wise folds in both nuScenes and Argoverse 2. Given the varied attributes of each city, the method's performance is sensitive to the composition of training and validation sets. As such, these city-wise folds ought to be utilized for cross-validation, where the average performance across different folds serves as the performance measure.

	nuScenes		Argoverse 2		
Set	A	B	A	B	C
Train	Boston, Onenorth	Boston, Queenstown, Holland Village	Miami, Pittsburgh	Miami, Rest	Pittsburgh, Rest
Val	Queenstown, Holland Village	Onenorth	Rest	Pittsburgh	Miami

Table 3. Far Extrapolation splits for nuScenes and Argoverse 2 where the folds are approximately similarly sized.

4. Experiments

To demonstrate the geographical data leakage problem with the original splits of nuScenes and Argoverse 2, we evaluate the performance of state-of-the-art online mapping methods on both the original and proposed geographically disjoint data splits. Additionally, we re-validate studies performed in previous works.

Unless specified, no modifications have been made to the configurations of the evaluated methods, and we direct readers to the respective papers for specific training details. Additionally, the performance is measured using standard practice in the respective field, *i.e.*, Intersection over Union (IoU) for segmentation-based methods and mean average precision (mAP) [16] for vector-based methods. For the latter, the average precision AP_τ is calculated through thresholding the Chamfer distance between matched prediction/ground truth-pairs for the thresholds $\tau \in T, T = \{0.5, 1.0, 1.5\}$ to get

$$\text{mAP} = \frac{1}{|T|} \sum_{\tau \in T} \text{AP}_\tau. \tag{1}$$

We report mAP for individual object classes and their mean.

4.1. Data Leakage Effects

To investigate the effects of data leakage across data partitions we train several vector- and segmentation-based methods on both the Original and the geographically disjoint Near Extrapolation splits. The results for vector-based

	Model	Sensor	Backbone	Lifting	Decoder	Split	Divider		Boundary		Crossing		Mean	
							Val	Test	Val	Test	Val	Test	Val	Test
nuScenes	VectorMapNet [20]	Camera	Resnet50	IPM	ART	Orig	48.9	47.9	40.9	63.8	39.8	52.8	43.2	54.8
						Near	13.5	17.3	14.9	21.6	13.7	15.7	14.0	18.2
	MapTR [18]	Camera	Resnet18	GKT	DETR	Orig	38.0	52.2	37.2	46.0	24.2	28.9	33.1	42.4
						Near	10.6	12.1	13.7	20.1	5.1	1.0	9.8	11.1
	MapTR [18]	Camera	Resnet50	GKT	DETR	Orig	51.0	65.8	52.6	60.5	43.0	53.3	48.8	59.9
						Near	16.0	19.9	26.7	33.3	14.4	5.9	19.0	19.7
	MapTRv2 [19]	Camera	Resnet50	LSS	DETR	Orig	61.8	77.3	63.7	70.9	59.1	69.8	61.5	72.7
						Near	20.9	23.4	32.6	40.5	26.5	14.8	26.7	26.2
	MapTRv2 [19]	Camera Lidar	Resnet50 SECOND	LSS	DETR	Orig	54.9	70.5	55.1	63.9	51.9	63.1	54.0	65.8
						Near	15.1	18.0	27.4	35.2	17.4	7.0	20.0	20.1
	StreamMapNet [39]	Camera	Resnet50	BEVFormer	DETR	Orig	64.5	79.3	62.3	69.5	61.0	74.5	62.6	74.4
						Near	23.0	22.6	29.5	35.2	25.8	26.0	26.1	27.9
Argoverse 2	VectorMapNet [20] 2D	Camera	Resnet50	IPM	ART	Orig	51.9	46.8	42.1	40.7	38.0	38.7	44.0	42.0
						Near	39.8	35.0	31.5	32.4	26.8	31.3	32.7	32.9
	MapTR [18] 2D	Camera	Resnet50	GKT	DETR	Orig	64.0	62.8	63.2	61.0	63.7	62.4	63.6	62.1
						Near	50.0	45.2	47.5	48.3	46.6	50.9	48.0	48.2
	MapTRv2 [19] 2D	Camera	Resnet50	LSS	DETR	Orig	71.7	68.9	67.0	63.8	64.5	63.1	67.7	65.3
						Near	58.4	56.6	51.3	53.5	49.7	55.6	53.1	55.2
	MapTRv2 [19] 3D	Camera	Resnet50	LSS	DETR	Orig	68.7	66.0	64.3	61.7	59.6	58.7	64.2	62.1
						Near	56.2	55.0	47.8	51.0	46.2	51.8	50.1	52.6
	StreamMapNet [39] 2D	Camera	Resnet50	BEVFormer	DETR	Orig	58.3	56.6	63.9	62.9	62.7	63.1	61.7	60.8
						Near	52.7	47.9	50.0	54.8	49.4	55.2	50.7	52.6

Table 4. mAP comparison for methods trained on Original (Orig) and Near Extrapolation (Near) splits. All methods show a significant performance drop when trained and evaluated on Near. Autoregressive Transformer [ART], Object Detection with Transformer [DETR].

methods are reported in Tab. 4. All evaluated methods see a significant performance drop when using geographically disjoint splits compared to the Original splits. The average performance decrement is more than 35 mAP and 12 mAP for nuScenes and Argoverse 2, respectively. Moreover, the effect is consistent over all lifting methods, sensor modalities, and decoders, but the ranking among the evaluated methods remains. The performance drop also remains consistent when adding lidar or considering the 3D geometry of the online map.

The best-performing method (MapTRv2) on nuScenes using images as input drops from an mAP of 72.7 to just 26.2, showcasing a difference of 46.5 mAP, when trained and evaluated appropriately. The drop is less pronounced, although still significant, on Argoverse 2, decreasing from 65.3 to 55.2 mAP. In general, the impact of the split is particularly distinct for methods trained on nuScenes. In light of these findings, we conclude that the smaller size of nuScenes, although convenient, makes it inadequate for training current online mapping methods. Moreover, although Argoverse 2 has more samples, it is somewhat surprising that algorithms trained on it still exhibit substantially improved generalization ability considering that the *intra-set* overlap is also larger. We hypothesize that, despite the *intra-set* overlap being greater, this overlap does not hinder training; instead, it possibly functions as natural and beneficial data augmentation similarly as synthetic augmentations have shown to be highly useful for image classification and object detection tasks [8].

To illustrate the significance of these numbers, a qualitative example is provided in Fig. 4, comparing MapTR [18] predictions on a validation sample when trained on the Original and the Near Extrapolation nuScenes splits. Studying the figure carefully, we can see that the model trained on the original split accurately predicts the road edge (green line) even for areas completely occluded in the images, highlighted by the teal box. The method also predicts the lane divider (yellow line) on the opposing road through the trucks and barriers, as highlighted in the pink box. Given their single-shot nature and lack of consideration for previous sensor data, it is unreasonable that they can accurately predict road structures that are not visible or clearly indicated by other structures in the current view. The method appears to learn to localize validation and test samples within the provided training map. This is not unique to this particular method or example, but rather a consequence of the overlap between train and evaluation sets. More qualitative examples are provided in Appendix D.

For segmentation-based methods, Tab. 5 shows reduced performance when evaluated on geographically disjoint data. This suggests the impact extends beyond vectorized methods. Here, HDMapNet is kept the same as from the original paper and the other lifting techniques, namely Inverse Perspective Mapping (IPM), Cross-View Transformer (CVT) and Geometry-Guided Kernel Transformer (GKT) have been altered to predict the classes we are interested in. Architectural details such as the image feature extraction backbone, Efficientnet-b4 [35], and segmentation decoder from SimpleBEV [13] are kept the same for all lifters for fair comparison. As for the vector-based online mapping

Figure 4. Predictions from MapTR [18] trained on Original and Geographical splits along with the ground truth. Yellow lines denote (lane) dividers, green (road) boundaries, and blue pedestrian crossings. Note that, when trained on the Original split, the branch to the parallel road on the left (teal box) is not visible in any image, yet appears in the predicted map. Also, the divider on the opposing road to the right (pink box) is predicted very well. When training on geographically split data (here Near Extrapolation), this method fails to predict these.

	Model	Split	Divider		Boundary		Crossing		Mean	
			Val	Test	Val	Test	Val	Test	Val	Test
nuScenes	GKT	Orig.	25.8	25.4	25.6	22.7	6.2	5.1	19.2	17.7
		Near.	12.5	17.9	12.6	16.9	1.4	1.9	8.8	12.3
	CVT	Orig.	30.9	30.1	30.5	25.5	11.7	7.6	24.4	21.1
		Near.	16.9	11.6	17.0	10.3	4.5	1.1	12.8	7.7
	IPM	Orig.	46.8	52.4	50.0	52.8	26.8	27.4	41.2	44.2
		Near.	29.6	29.4	36.2	33.6	16.2	9.7	27.4	24.2
	HDMapNet	Orig	38.3	47.5	35.5	41.2	20.1	27.6	31.3	38.8
		Near	8.6	24.0	22.1	25.4	10.6	14.1	17.1	21.2
Argoverse 2	GKT	Orig.	37.3	28.2	31.4	28.3	10.3	5.7	26.3	20.7
		Near.	32.7	29.0	26.2	23.2	8.7	1.5	22.5	17.9
	CVT	Orig.	40.1	29.4	32.0	29.3	12.1	7.1	28.4	21.9
		Near.	35.1	31.7	26.5	28.5	10.9	1.4	24.2	20.5
	IPM	Orig.	58.5	45.7	50.6	50.1	33.4	32.4	47.5	42.7
		Near.	50.5	39.1	44.1	45.2	29.7	28.8	41.5	37.7

Table 5. Map segmentation on the datasets, evaluating performance with IoU. The performance drops for all methods using the Near. versus Orig.. As for the vector-based OME, the drop is larger on nuScenes than Argoverse 2.

methods, the drop is larger on nuScenes than Argoverse 2 and the ranking among methods remains largely unchanged.

Overall, the Near Extrapolation split yields a more consistent performance between the validation and test sets compared to the Original split. Suggesting a balanced distribution across sets and facilitates drawing reliable conclusions about hyperparameter choices. Ensuring that insights gained from the validation set generalize well to the test set.

4.2. Far Extrapolation Cross-validation

We perform cross-validation using multiple folds of city-wise data partitioning to evaluate the performance under increased distribution shifts. Tab. 6 shows the vector-based methods' performance for both nuScenes and Argoverse 2 using the Far Extrapolation splits. The performance drops even further using this split, emphasizing the difficulty current methods experience with extrapolating outside the training distribution.

	Model	Split	Divider	Boundary	Crossing	Mean	CV
nuScenes	VectorMapNet	A	7.6	8.4	5.9	7.3	8.7
		B	11.9	12.2	6.1	10.1	
	MapTR	A	12.9	21.1	11.1	15.0	14.9
		B	14.1	24.6	5.9	14.9	
	MapTRv2	A	18.6	27.9	18.9	21.8	21.4
		B	22.4	27.0	13.3	20.9	
	StreamMapNet	A	16.4	22.7	18.7	19.3	21.3
		B	21.7	30.6	17.4	23.2	
Argoverse 2	VectorMapNet	A	16.5	19.0	16.5	21.1	24.0
		B	29.4	26.0	20.5	25.3	
		C	28.0	27.1	22.1	25.7	
	MapTR	A	41.7	37.3	34.7	37.9	41.8
		B	47.5	45.9	40.2	44.5	
		C	41.2	44.6	43.3	43.0	
	MapTRv2	A	42.2	41.9	37.4	40.5	45.5
		B	53.4	50.9	42.0	48.8	
		C	50.3	48.6	42.6	47.2	
	StreamMapNet	A	43.4	43.3	40.2	42.3	46.6
		B	51.0	52.5	46.7	50.1	
		C	42.7	50.1	49.4	47.4	

Table 6. Vector-based methods' mAP on the Far Extrapolation folds and their corresponding cross-validation mean (CV).

4.3. Sample Density

We investigate the effect of the training set's sample density for both datasets. As discussed in Sec. 2.2 only the key-frame samples are typically used when training methods on nuScenes. We leverage the fact that *all* samples have the vehicle poses required to extract the ground truth from the HD map and are able to use 4 times as many samples for training. It is also forthright to downsample Argoverse 2 to the desired sample density by selecting every fourth sample, effectively simulating the sample density of nuScenes.

The training schedule is adjusted such that the total number of optimizer steps is similar between the sparsely and densely sampled data. Tab. 7 reports the result for MapTRv2. For nuScenes there is a distinguished increase in performance on the original split, achieving up to a 24.9 mAP improvement on the validation set by utilizing the densely sampled data. However, the method's performance using the Near Extrapolation split sees only marginal improvement (max increase of 0.7 mAP). Argoverse 2 sees smaller differences between the sample densities.

	Split	Train Sampling	Divider		Boundary		Crossing		Mean	
			Val	Test	Val	Test	Val	Test	Val	Test
nuScenes	Orig.	Sparse	61.8	77.3	63.7	70.9	59.1	69.8	61.5	72.7
		Dense	90.3	89.7	84.4	84.5	86.4	88.3	86.4	87.5
	Near	Sparse	20.9	23.4	32.6	40.5	26.5	14.8	26.7	26.2
		Dense	21.7	24.1	34.1	40.9	25.7	15.9	27.2	26.9
Argo 2	Orig.	Sparse	67.7	64.9	63.6	59.8	58.8	57.1	63.4	60.6
		Dense	71.7	68.9	67.0	63.8	64.5	63.1	67.7	65.3
	Near	Sparse	55.4	54.0	48.2	52.5	44.8	51.5	49.5	52.6
		Dense	58.4	56.6	51.3	53.5	49.7	55.6	53.1	55.2

Table 7. MapTRv2 mAP trained on varying dataset density. For nuScenes, dense sampling boosts performance by up to 24.9 on the Orig., while Near only sees minor improvements.

4.4. Re-validation

As original works validate design choices and hyperparameters on poorly separated data, it is highly relevant to revisit these results to check applicability to proposed split. Though not covering all prior tests, we highlight some interesting observations on the Near Extrapolation split. Hyperparameter-search shows minor differences, see Appendix C, while new conclusions emerge regarding lifting method and auxiliary tasks.

Lifting methods In [18], an ablation study investigates the impact of various lifters for MapTR, with GKT yielding the best results. However, upon re-running this test (see Tab. 8), we observe a contradiction to the previous findings. The BEVFormer lifter slightly outperforms GKT. Nonetheless, the differences between the lifters are marginal, making it challenging to determine the superiority of any specific lifter.

Auxiliary tasks For MapTRv2 [19], we re-run the ablation studies on the proposed auxiliary tasks in Tab. 9. For nuScenes, we note that, in contrast to conclusions based on the original split, the addition of depth supervision does not yield a significant performance boost. Additionally, one can infer that it is only when all auxiliary tasks are combined that the improvement becomes apparent. However, the effects of the additional tasks are smaller than initially concluded when training on the original split. Considering Argoverse 2, there are bigger differences between the performance among the auxiliary tasks. Similarly to the re-validation on nuScenes, the effectiveness of, *e.g.*, depth supervision is not as striking as previously advertised.

5. Conclusion

We propose and employ geographically disjoint splits of the most used datasets, revealing that the performance of state-of-the-art online mapping methods is significantly lower than previously reported. We argue that these splits offer a more accurate measure of how well online mapping methods generalize to new geographic areas. While the Near Extrapolation split acts as a drop-in replacement to the original splits, we urge the community to target the Far Extrapolation setting moving forward.

Even though performance numbers have decreased for all methods with these splits, the ranking between methods remains largely the same. The performance disparity is more pronounced on nuScenes than Argoverse 2. However, our follow-up re-validation experiments have revealed new insights, diverging from conclusions based on the original split. Notably, the impact of the lifting method and the support from auxiliary tasks, *e.g.* depth supervision, on performance appears less substantial or follows a different trajectory than initially perceived.

In summary, online mapping remains a formidable challenge, and to make substantial progress, we must anchor our conclusions in fair evaluations based on clean data splits. We look forward to what innovations will come from the improved evaluation ability with the release of our geographically disjoint data splits.

	Lifting	Split	Divider	Boundary	Crossing	Mean
nuScenes	GKT	Orig	51.0	52.6	43.0	48.8
		Near	16.0	26.7	14.4	19.0
	BEVFormer	Orig	49.7	53.5	40.2	47.8
		Near	16.2	28.2	17.5	20.6
	LSS	Orig	52.1	52.4	45.4	50.0
		Near	17.3	27.7	18.0	21.0
Argoverse 2	GKT	Orig	64.0	63.2	63.7	63.6
		Near	50.0	47.5	46.6	48.0
	BEVFormer	Orig	63.7	63.8	63.0	63.5
		Near	49.5	47.2	46.3	47.7
	LSS	Orig	64.8	65.2	61.9	64.0
		Near	49.7	47.3	45.1	47.3

Table 8. Validation mAP for lifting methods in MapTR. Marginal differences between the lifters make it challenging to establish the superiority of any particular method.

Depth	Seg^{PV}	Seg^{BEV}	nuScenes		Argoverse 2
			Orig.	Geo.	Geo.
			56.6	25.3	48.8
✓			59.8	25.9	48.8
✓	✓		60.5	26.1	50.6
✓		✓	61.0	25.9	50.9
	✓	✓	59.2	25.5	51.5
✓	✓	✓	**61.5**	**26.7**	**53.1**

Table 9. Validation mAP for auxiliary tasks in MapTRv2. nuScenes Orig. numbers are from [19]. Nearyields a smaller performance boost of auxiliary tasks. For Argoverse 2, the differences are greater, but inconsistent with the result on Orig..

Acknowledgements: This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Computational resources were provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at NSC Berzelius and C3SE Alvis partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

References

[1] Mina Alibeigi, William Ljungbergh, Adam Tonderski, Georg Hess, Adam Lilja, Carl Lindstrom, Daria Motorniuk, Junsheng Fu, Jenny Widahl, and Christoffer Petersson. Zenseact open dataset: A large-scale and diverse multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20178–20188, 2023. 1, 4

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020. 1, 2, 3

[3] Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Structured bird's-eye-view traffic scene understanding from onboard images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15661–15670, 2021. 3

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3

[5] Athanasios Chalvatzaras, Ioannis Pratikakis, and Angelos A Amanatiadis. A survey on map-based localization techniques for autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 8(2):1574–1596, 2022. 1

[6] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3, 4

[7] Shaoyu Chen, Tianheng Cheng, Xinggang Wang, Wenming Meng, Qian Zhang, and Wenyu Liu. Efficient and robust 2d-to-bev representation learning via geometry-guided kernel transformer, 2022. 2, 3

[8] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020. 6

[9] Hao Dong, Xianjing Zhang, Jintao Xu, Rui Ai, Weihao Gu, Huimin Lu, Juho Kannala, and Xieyuanli Chen. Superfusion: Multilevel lidar-camera fusion for long-range hd map generation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 2, 3

[10] P. Ploton et al. Spatial validation reveals poor predictive performance of large-scale ecological mapping models. In *Nature Communications, vol. 11, no. 1, p. 4540*, 2020. 4

[11] Hao Feng, Yongcheng Wang, Zheng Li, Ning Zhang, Yuxi Zhang, and Yunxiao Gao. Information leakage in deep learning-based hyperspectral image classification: A survey. *Remote Sensing*, 15(15), 2023. 4

[12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 1

[13] Adam W Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. Simple-bev: What really matters for multi-sensor bev perception? In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2759–2765. IEEE, 2023. 6

[14] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pages 533–549. Springer, 2022. 2, 3

[15] Yanqin Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yu-Gang Jiang. Polarformer: Multi-camera 3d object detection with polar transformer. In *Proceedings of the AAAI conference on Artificial Intelligence*, pages 1042–1050, 2023.

[16] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4628–4634. IEEE, 2022. 2, 3, 4, 5

[17] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, pages 1–18. Springer, 2022. 2, 3

[18] 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 *The Eleventh International Conference on Learning Representations*, 2023. 2, 3, 4, 6, 7, 8

[19] Bencheng Liao, Shaoyu Chen, Yunchi Zhang, Bo Jiang, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Maptrv2: An end-to-end framework for online vectorized hd map construction, 2023. 1, 3, 6, 8

[20] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *International Conference on Machine Learning*, pages 22352–22369. PMLR, 2023. 2, 3, 4, 6

[21] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 2774–2781. IEEE, 2023. 2, 3

[22] Hanspeter A Mallot, Heinrich H Bülthoff, JJ Little, and Stefan Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological cybernetics*, 64(3):177–185, 1991. 3

[23] Lang Peng, Zhirong Chen, Zhangjie Fu, Pengpeng Liang, and Erkang Cheng. Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5935–5943, 2023. 2, 3

[24] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020. 2, 3, 1

[25] Zequn Qin, Jingyu Chen, Chao Chen, Xiaozhi Chen, and Xi Li. Unifusion: Unified multi-view fusion transformer for spatial-temporal representation in bird's-eye-view. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8690–8699, 2023. 3, 4

[26] David R. Roberts, Volker Bahn, Simone Ciuti, Mark S. Boyce, Jane Elith, Gurutzeta Guillera-Arroita, Severin Hauenstein, José J. Lahoz-Monfort, Boris Schröder, Wilfried Thuiller, David I. Warton, Brendan A. Wintle, Florian Hartig, and Carsten F. Dormann. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40(8):913–929, 2017. 4

[27] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11138–11147, 2020. 2, 3, 4

[28] E. Rolf. Evaluation challenges for geospatial ml. In *arXiv preprint arXiv:2303.18087v1*, 2023. 4

[29] Avishkar Saha, Oscar Mendez, Chris Russell, and Richard Bowden. Translating images into maps. In *2022 International conference on robotics and automation (ICRA)*, pages 9200–9206. IEEE, 2022. 3

[30] Juyeb Shin, Francois Rameau, Hyeonjun Jeong, and Dongsuk Kum. Instagram: Instance-level graph modeling for vectorized hd map learning. *arXiv preprint arXiv:2301.04470*, 2023. 2, 3, 4

[31] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Peter Kontschieder, and Elisa Ricci. Are we missing confidence in pseudo-lidar methods for monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3225–3233, 2021. 4

[32] Kai Sun, Yingjie Hu, Gaurish Lakhanpal, and Ryan Zhenqi Zhou. *Spatial cross-validation for GeoAI*. Handbook of Geospatial Artificial Intelligence, CRC Press 201-214, 2021. 4

[33] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 1, 3

[34] Iulian Emil Tampu, Anders Eklund, and Neda Haj-Hosseini. Inflation of test accuracy due to data leakage in deep learning-based classification of oct images. *Scientific Data*, 9(1), 2022. 4

[35] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 6

[36] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks)*, 2021. 1, 2, 3

[37] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M Alvarez. Mˆ 2bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. *arXiv preprint arXiv:2204.05088*, 2022. 2, 3

[38] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 1

[39] Tianyuan Yuan, Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. Streammapnet: Streaming mapping network for vectorized online hd map construction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7356–7365, 2024. 3, 4, 6

[40] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv preprint arXiv:2205.09743*, 2022. 2, 3

[41] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13760–13769, 2022. 3

Localization Is All You Evaluate: Data Leakage in Online Mapping Datasets and How to Fix It

Appendix

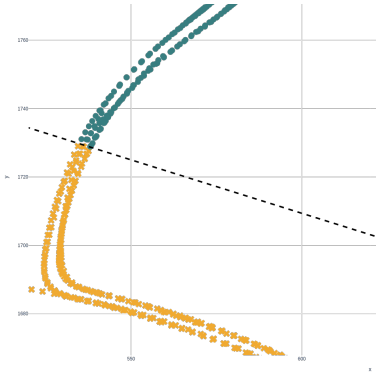


Figure 5. In this example from Singapore Queenstown, the individual samples from a few sequences are divided into training (green) and validation (orange) according to the cut-off border. Some samples from a sequence are put in the training set, whereas the remaining are put in the validation set. The samples close to the dotted black cut-off line are the remaining possible data-leakage samples when using our proposed Near Extrapolation split.

A. Partially Overlapping Maps

Splitting nuScenes for Near Extrapolation on a sequence level requires grouping large areas with similar zone classes together and putting them in a single set, as seen in [24]. This is due to the entangled nature of the sequences where many partially overlap. Instead, we assign each sample individually to a set when a sequence straddles the boundary between two sets (*e.g.* train and val in Fig. 5). We divide the sequence at the boundary, creating two separate partial sequences each with preserved temporal consistency. This maintains the usefulness for object detection and keeps the possibility of using the data for temporal fusion, where having consecutive samples is important. We have kept the number of sequences being cut into multiple parts as low as possible, making the cuts, when necessary, across the road's driving direction.

The sequences in the Argoverse 2 dataset are more spread out compared to nuScenes, and a balanced sequence-wise split is possible to obtain. There is thus no impact on usability for object detection, object tracking, and other temporal fusion applications for the Argoverse 2 split.

Splitting the data geographically ensures that there is no overlap in poses between the different sets. However, as online mapping methods typically predict 30m in front and to the rear there will still be some overlap in the ground truth maps among the samples close to the cut-off border.

Split	HDMapNet		VectorMapNet		MapTR		MapTRv2	
	Val	Test	Val	Test	Val	Test	Val	Test
Near	17.1	21.2	14.0	18.2	19.0	19.7	26.7	26.2
Near$\notin$	17.0	21.4	14.5	18.3	19.0	19.6	26.5	25.8

Table 10. Evaluating the predictions from validation and test sets in the nuScenes' Near Extrapolation split, where samples closer than 60m to a training sample have been removed (indicated by $\notin$). It can be seen that the impact on performance is negligible. Metrics are IoU for HDMapNet, and mAP for MapTRv2 and VectorMapNet.

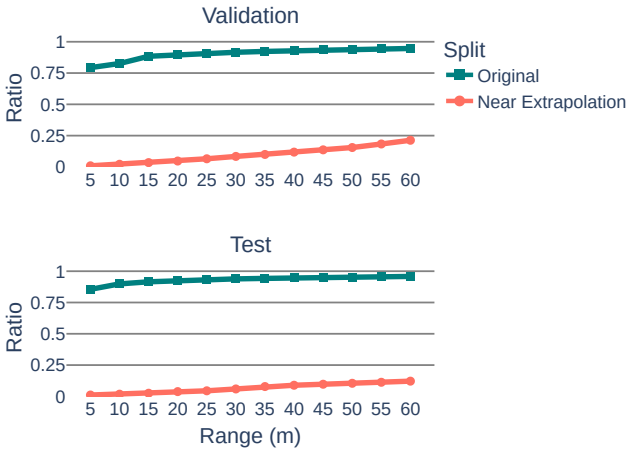


Figure 6. Ratios of validation and test samples within a certain range of training samples for nuScenes. The Geographically disjoint Near Extrapolation split has negligible overlap compared to the, greatly overlapping, Original split.

To see the effects of the remaining overlap in the geographical split of nuScenes we run experiments where the validation and test samples closer than 60 m to a training sample have been filtered out. Tab. 10 demonstrates that these samples have a negligible impact on performance. Furthermore, Fig. 6 shows how the ratio of validation and test samples that are close to a training sample changes with range. For completeness Fig. 7 displays the same information on Argoverse 2.

B. Additional Data Attributes

In this section, we further display the splitting, the number of samples in discretized maps, and different zone classes (*e.g.* residential, commercial, and industrial).

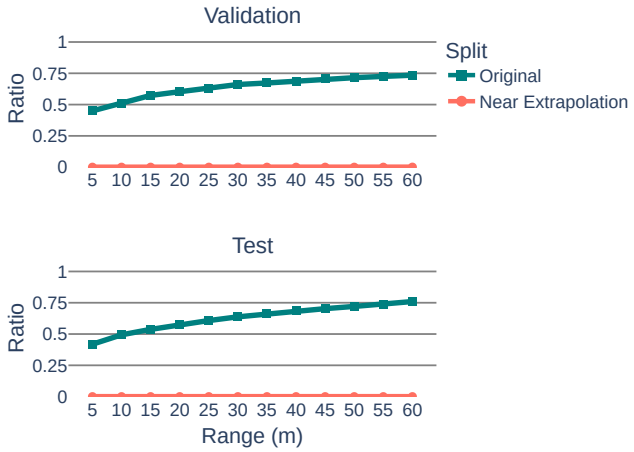


Figure 7. Ratios of validation and test samples within a certain range of training samples for Argoverse 2. The Geographically disjoint Near Extrapolation split has no overlap compared to the, greatly overlapping, Original split.

Figure 8. Ratios of weather conditions (clear and rain) as well as time of day (day and night) on the nuScenes dataset. The black dashed lines are the respective ratios over the full dataset.

nuScenes Fig. 8 details that the Near Extrapolation split is balanced across all attributes. This allows for conducting experiments and drawing conclusions on a well-defined dataset. Further, Fig. 10 depicts example images and their position on the map for Boston Seaport. The industrial zones in the south and south-eastern areas have different attributes, *e.g.* type of buildings, lane widths, number of lanes, and frequency of pedestrian crossings, than the commercial and residential zones in the north-western part. It is thus important that these zones are represented in all sets for a fair evaluation of trained methods. Fig. 11 showcases the regions where samples are allocated in each set for all cities. Each set incorporates regions from different parts of the cities to promote diversity. The heatmaps in Fig. 12 depict the distribution of samples within each 60m cell. One can, for instance, observe a concentration of samples in crossings.

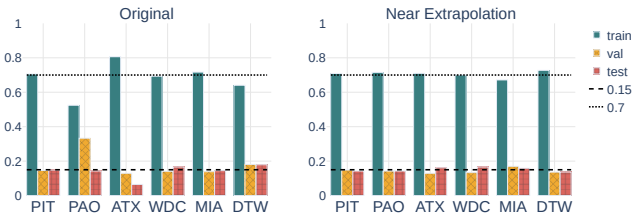


Figure 9. Inter-set city distribution for the Original and Geographically split data of Argoverse 2. The dotted and dashed lines represent the 70% and 15% target ratios respectively.

Argoverse 2 As discussed in Sec. 3.1 it is possible to split Argoverse 2 on a sequence level while preserving zone class diversity. Fig. 9 illustrates the distribution of the number of samples in each city. Fig. 13 further highlights the significance of diverse city areas in all sets by presenting a collection of images from various locations in Washington DC. The downtown area in the southwest exhibits different road characteristics from the sub-urban areas in the north and east. Fig. 14 illustrates how the complete set of city maps are split, ensuring a diverse selection of areas in each set. Heatmaps in Fig. 15 represent the distribution of samples within each 60m cell.

C. Extended Experiments

To further investigate the effects of data splits and the amount of data, we perform additional experiments.

Far Extrapolation We also train and evaluate the segmentation-based methods with city-wise folds. Tab. 11 reports the performance for the folds in the Far Extrapolation split and their cross-validation mean. The performance on these folds are, similarly to the Near Extrapolation splits, lower than on the Original splits. For nuScenes the performance on Near Extrapolation splits is already low, and the performance on the Far Extrapolation folds are on par. For Argoverse 2 the city-wise folds are performing worse than the Near Extrapolation split's validation set, but similar to the test set. This results in the cross-validation mean being consistently lower than the mean of the validation and test performance of the Near Extrapolation splits.

Training set extension To explore how the amount of data affects the performance we extend the training set with the validation samples, effectively increasing the training set with 20%. Tab. 12 shows a boost in the test performance for both segmentation- and vector-based methods. The impact is greater on nuScenes, but Argoverse 2 also benefits from the added data, indicating that more extensive datasets are necessary for learning online mapping. For instance, the extra data has a higher impact for MapTR on Argoverse 2,

	Model	Split	Divider	Boundary	Crossing	Mean	CV
nuScenes	GKT	A	10.6	14.2	0.8	8.5	9.9
		B	14.7	17.2	1.6	11.2	
	CVT	A	13.1	14.1	2.2	9.8	10.7
		B	14.8	17.5	2.6	11.6	
	IPM	A	28.1	34.0	12.1	24.7	26.6
		B	33.6	38.8	13.0	28.5	
	HDMapNet	A	20.1	20.7	7.2	16.0	27.3
		B	24.2	24.4	6.9	18.5	
Argoverse 2	GKT	A	28.2	21.8	7.1	19.0	20.1
		B	30.8	25.8	6.8	21.1	
		C	29.5	23.7	6.9	20.0	
	CVT	A	29.0	21.9	9.5	20.1	20.8
		B	31.9	23.0	7.6	20.8	
		C	30.6	24.0	9.3	21.3	
	IPM	A	43.0	38.2	24.6	35.3	37.4
		B	48.6	43.3	25.8	39.2	
		C	45.1	41.6	26.8	37.8	

Table 11. Segmentation-based methods' IoU on the city-wise folds of the Far Extrapolation split and their corresponding cross-validation mean (CV).

	Model	Split	Divider	Boundary	Crossing	Mean
nuScenes	HDMapNet	Near	15.3	17.3	9.0	13.9
		Near∪	24.4	26.3	14.7	21.8
	VectorMapNet	Near	17.3	21.6	15.7	18.2
		Near∪	18.8	25.3	17.6	20.6
	MapTR	Near	19.9	33.3	5.9	19.7
		Near∪	21.9	36.1	6.8	21.6
	MapTRv2	Near	23.4	40.5	14.8	26.2
		Near∪	25.3	42.1	18.6	28.7
Argoverse 2	VectorMapNet	Near	35.0	32.4	31.3	32.9
		Near∪	37.5	33.1	32.7	34.4
	MapTR	Near	45.2	48.3	50.9	48.2
		Near∪	47.3	49.4	52.0	49.6
	MapTRv2 2D	Near	56.6	53.5	55.6	55.2
		Near∪	59.5	54.7	53.4	55.9

Table 12. Increasing training data by 15% using the union of training and validation samples, marked by ∪, improves test performance. IoU for HDMapNet and mAP for the other methods.

+1.4 mAP, than the choice of lifting method, +0.7 mAP. On nuScenes, the impact is greater, but also similarly large as using LSS in comparison to GKT for lifting, +1.9 and +2.0 respectively. The lifting methods are further discussed in Sec. 4.4 and shown in Tab. 8.

Hyperparameter-search For MapTRv2 on the Near Extrapolation split on nuScenes, we investigate various hyperparameters related to overfitting on the training set, *i.e.*, weight decay, learning rate, and training epochs. Interestingly, we can in Tab. 13 observe only minor differences and the parameters initially employed for training on the Original split seem equally effective for the geographically disjoint split.

D. Qualitative Results

nuScenes Fig. 16 portrays three examples with input images, the evaluation prediction, its ground truth, and the closest training sample. Despite not being captured from the exact same pose, these instances demonstrate striking similarities between the evaluation and closest training pose. This underscores that the method, having encountered the closest training sample during training, can achieve accurate predictions through memorization and retrieval of these examples at test time. Additional examples can be seen in the videos to be part of the project webpage.

		LR	
		$6e^{-4}$	$1e^{-4}$
WD	0.05	27.0	26.5
	0.10	26.7	27.2
	0.15	27.1	26.5

Table 13. MapTRv2 show robustness to different hyperparameters, learning rate (LR) and weight decay (WD) on nuScenes Near Extrapolation split.

Fig. 17 compares the predictions of a sample included in the test set of both the Original and Near Extrapolation splits. It demonstrates that a model trained on the Original split can predict dividers, boundaries, and pedestrian crossings occluded by vehicles in the opposing lane accurately. Thus making it tempting to speculate that the method has memorized this information. Furthermore, it shows that the model trained on geographically disjoint data only identifies the dividers near the ego vehicle. These dividers are visible in the images, but absent in the ground truth, indicating that the model has learned to generalize better.

Argoverse 2 In Fig. 18, we present three examples featuring input images, the evaluation prediction, its ground truth, and the closest training sample. While not being from the exact same pose, *e.g.* in the top example the closest training pose is slightly rotated, and in the bottom from an adjacent lane, it is still plausible for a method to achieve a high score on the test sample by memorizing the map and images from the training sample, and then recall and slightly shift and rotate that map at test time. Additional examples will be available on the project webpage.

Fig. 19 illustrate comparisons between predictions derived from a sample included in both the Original and Geographically disjoint splits' test set, along with the ground truth. Despite the inherent difficulty in predicting objects situated behind a truck on the left side, the model trained on the original split demonstrates commendable accuracy in its estimations. The model also effectively predicts the lane divider to the right of the ego vehicle, when not visibly present in the image but existing in the ground truth. It is worth noting that this may not be due to memorization, as the model could learn, e.g., consistent data annotations and hints from road dividers and road width to accurately predict this non-visible lane divider.

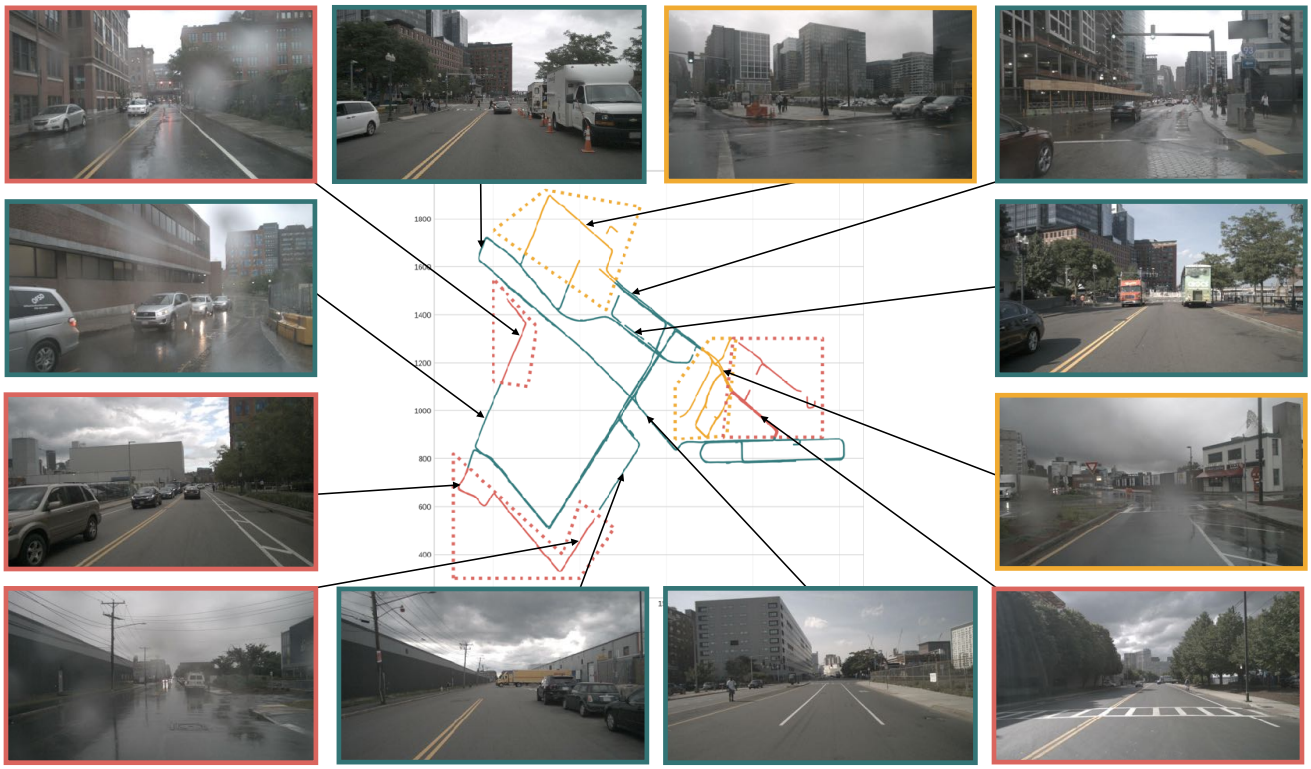
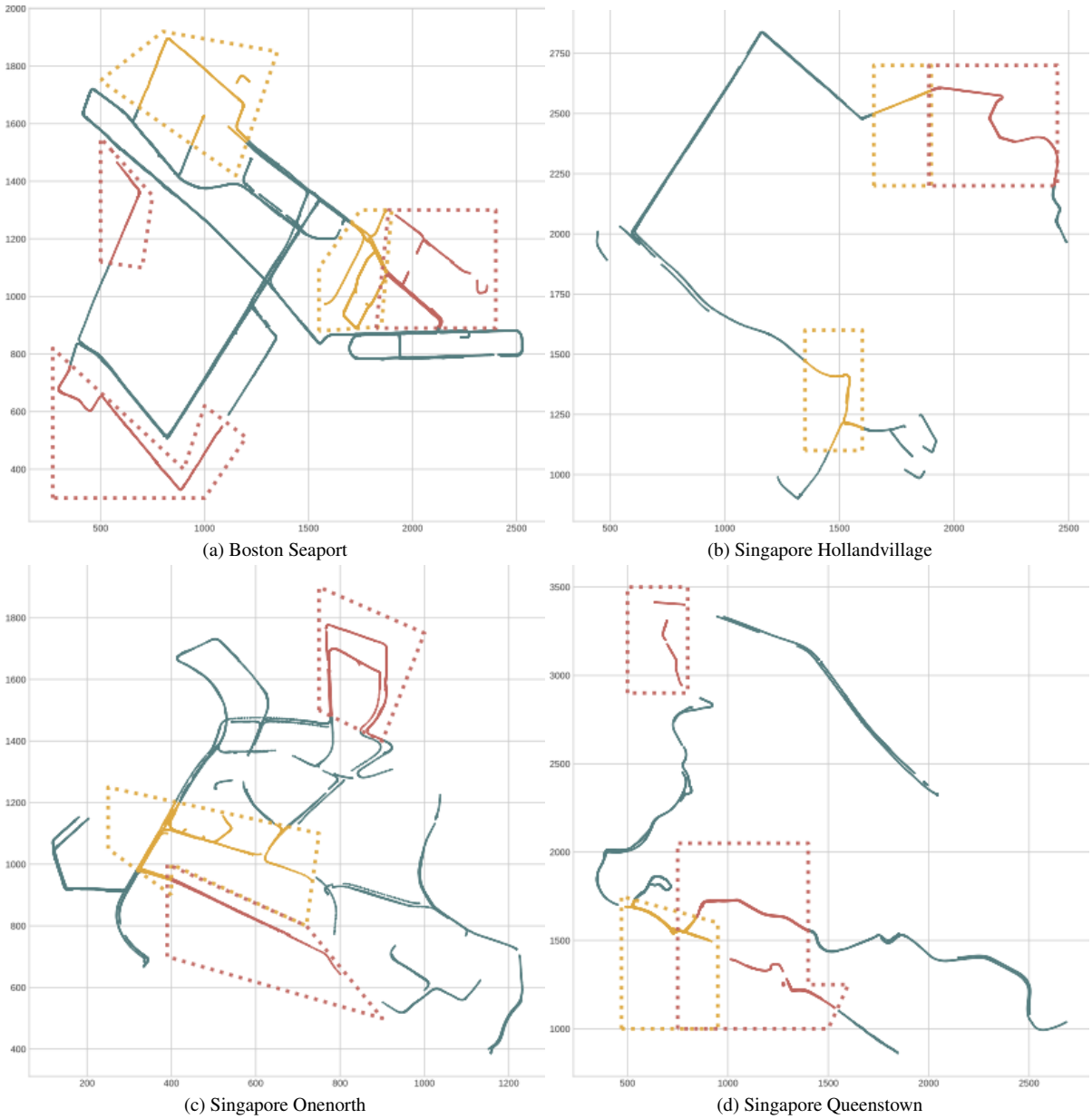


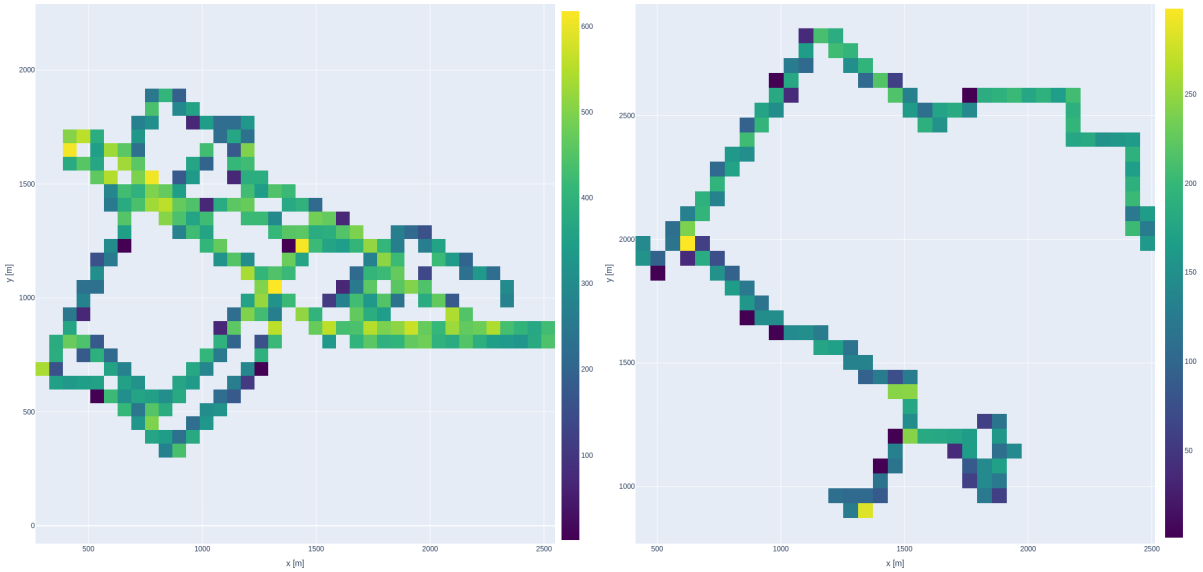
Figure 10. Selected poses from the Boston Seaport map in nuScenes dataset, with marked training (green), validation (blue), and test (red) poses according to the Near Extrapolation split. Dotted polygons mark the boundaries of the validation and test zones. To ensure diversity in zone types within each set, regions from various parts of the city are included. The industrial zones in the south and south-eastern areas have different attributes than the commercial and residential zones in the north-western part.



(a) Boston Seaport (b) Singapore Hollandvillage

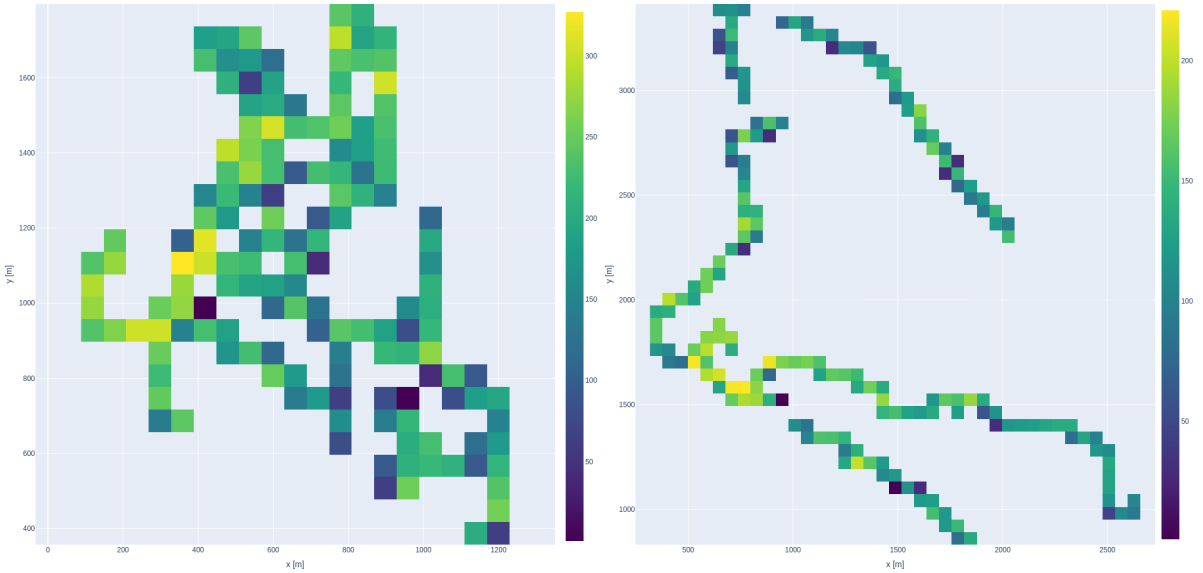
(c) Singapore Onenorth (d) Singapore Queenstown

Figure 11. Positions of samples in the nuScenes dataset, with the geographical areas of the Near Extrapolation split outlined by dotted polygons. Training, validation, and test sets are distinguished by green, orange, and red colors, respectively. Areas from various parts of the cities are present in each set.



(a) Boston Seaport

(b) Singapore Hollandvillage

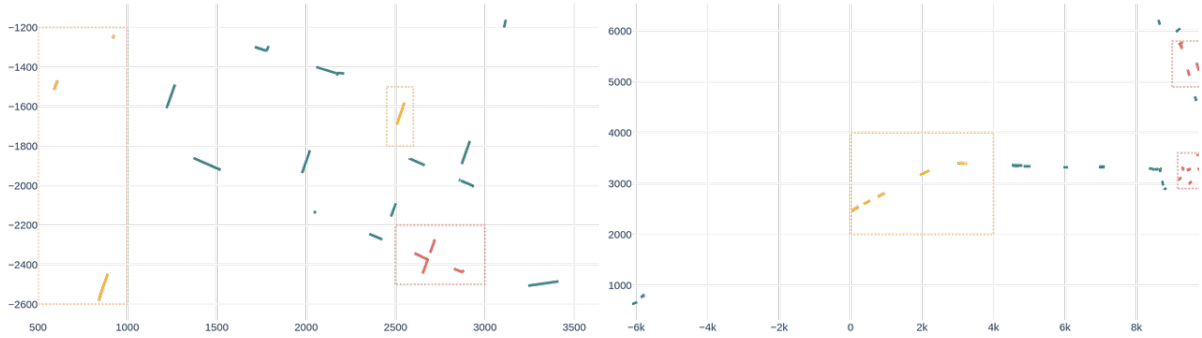


(c) Singapore Onenorth

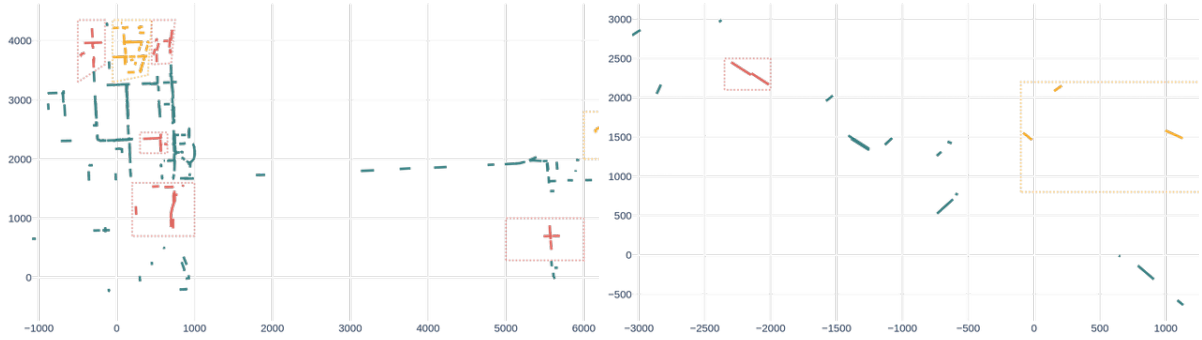
(d) Singapore Queenstown

Figure 12. Heatmaps depicting the distribution of samples within 60m cells in the nuScenes dataset, revealing a high amount of samples in many cells, especially concentrated within crossings.

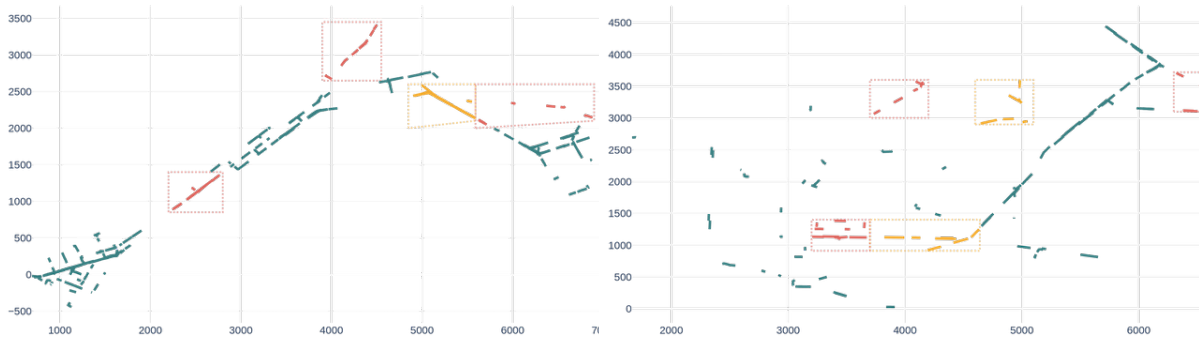
Figure 13. Samples from the Washington DC map in Argoverse 2 dataset, with marked training (green), validation (blue), and test (red) pose according to the Near Extrapolation split. Dotted polygons mark the boundaries of the validation and test zones. To enhance diversity in zone types within each set, regions from different parts of the city are incorporated. The downtown area in the southwest has different road characteristics from the sub-urban areas in the north and eastern parts.



(a) Austin (b) Detroit

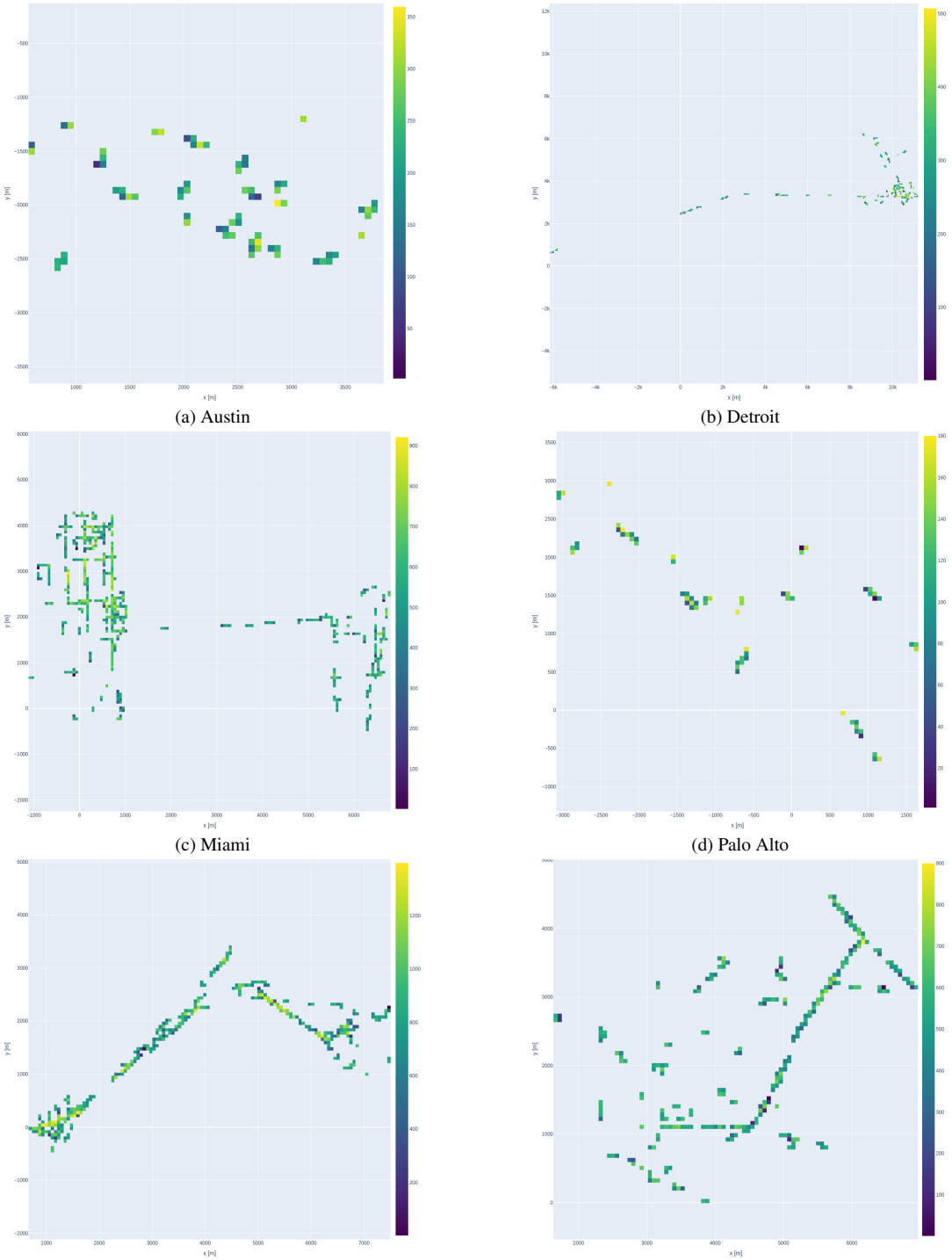


(c) Miami (d) Palo Alto



(e) Pittsburgh (f) Washington DC

Figure 14. Near Extrapolation. Positions of samples in the nuScenes dataset, with the geographical areas of the validation and test sets outlined by dotted polygons. Training, validation, and test sets are distinguished by green, orange, and red colors, respectively. Regions from different parts of the cities are present in each set.



(a) Austin (b) Detroit

(c) Miami (d) Palo Alto

(e) Pittsburgh (f) Washington DC

Figure 15. Heatmaps for the number of samples within 60m cells for Argoverse 2 dataset. Many cells contain a lot of samples, with the maximum number of samples in a single cell being 1398.

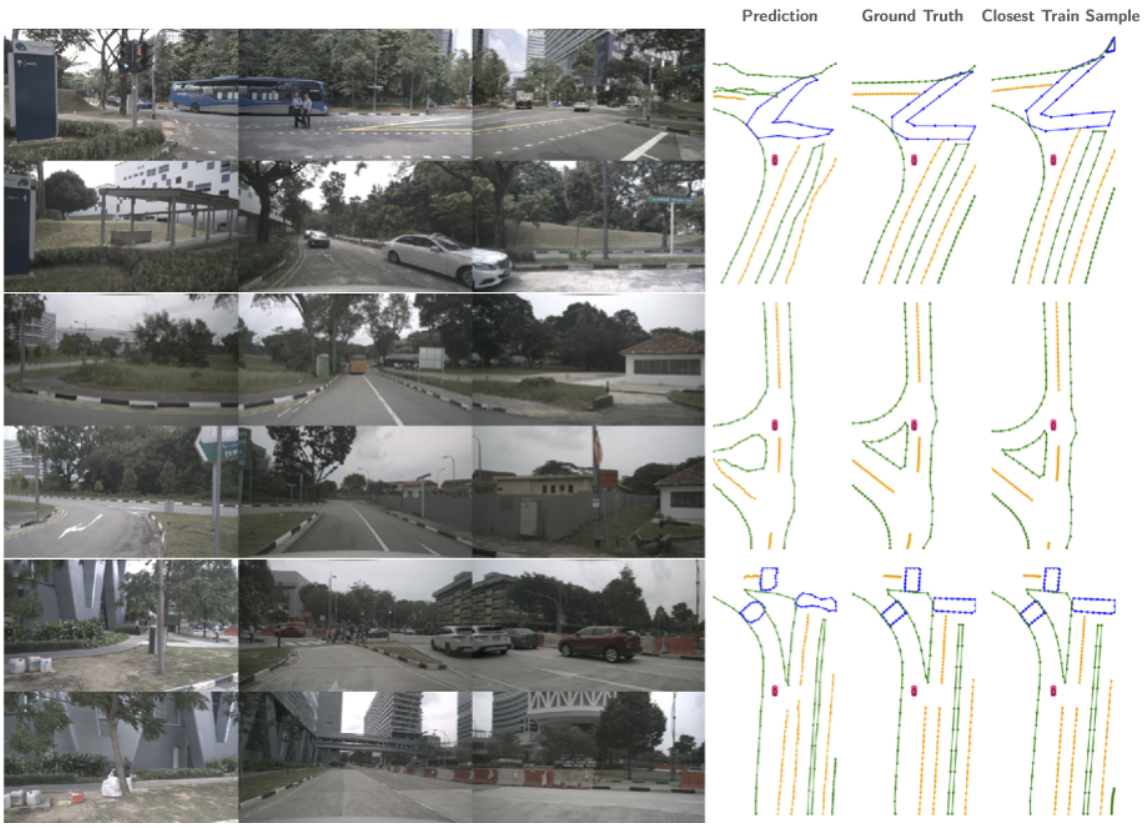


Figure 16. Multiple examples of validation or test prediction from MapTR on nuScenes, corresponding ground truth, and the closest training sample's ground truth. The close similarities between the closest training samples and the evaluation samples are evident in each example.

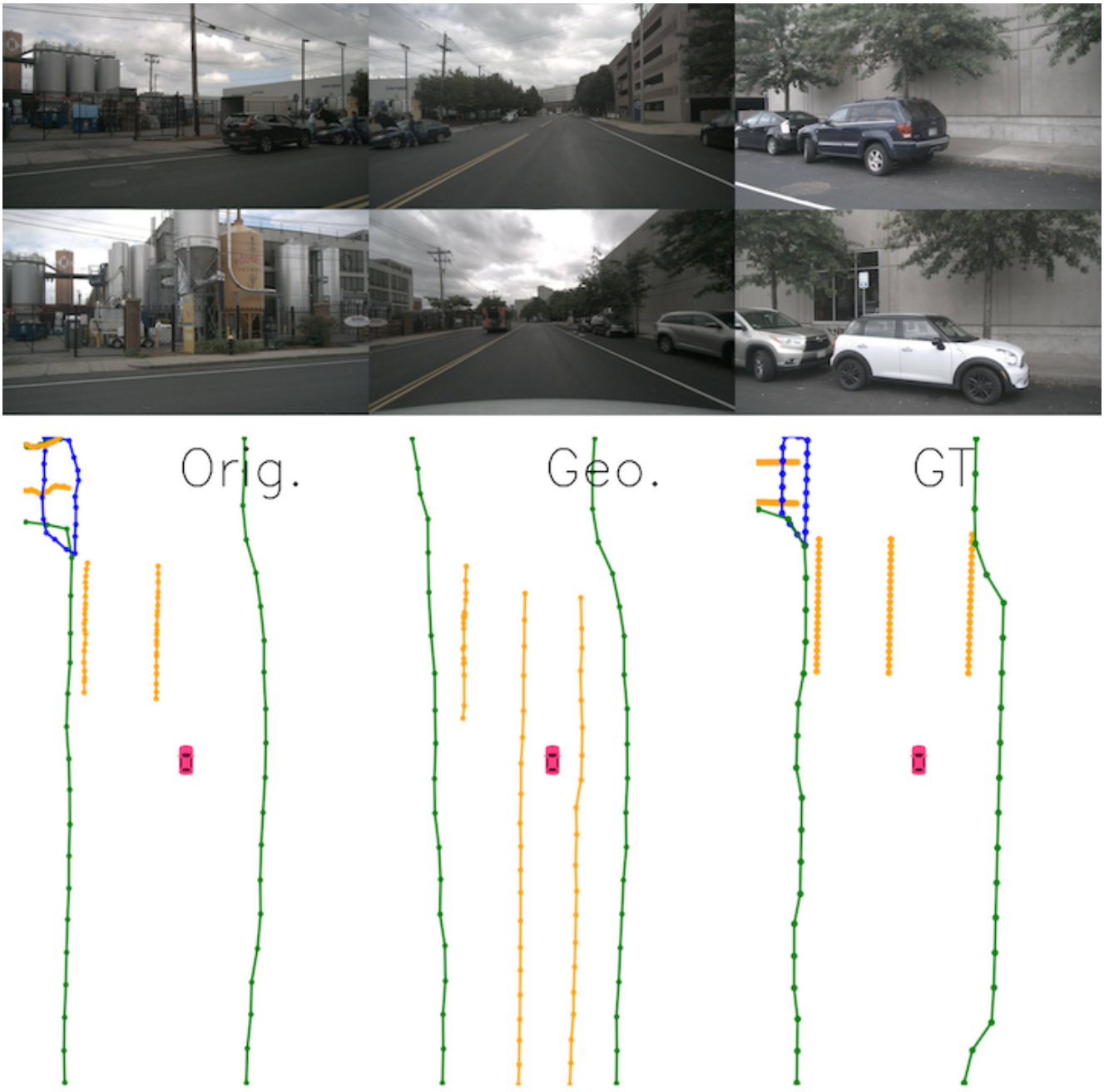


Figure 17. nuScenes test prediction from MapTR trained on Original (Orig.) and Geographically disjoint (Geo.), here Near Extrapolation, along with the ground truth (GT). Dividers, Boundaries, and Pedestrian crossings are visualized in orange, green, and blue respectively. Despite occlusion on the left side by opposing lane vehicles, the method trained on the original split accurately predicts them. In contrast, the model trained on geographically disjoint splits fails to detect them. On the other hand, the model trained on geographically disjoint split data successfully identifies dividers near the ego vehicle, even though they are absent in the ground truth.

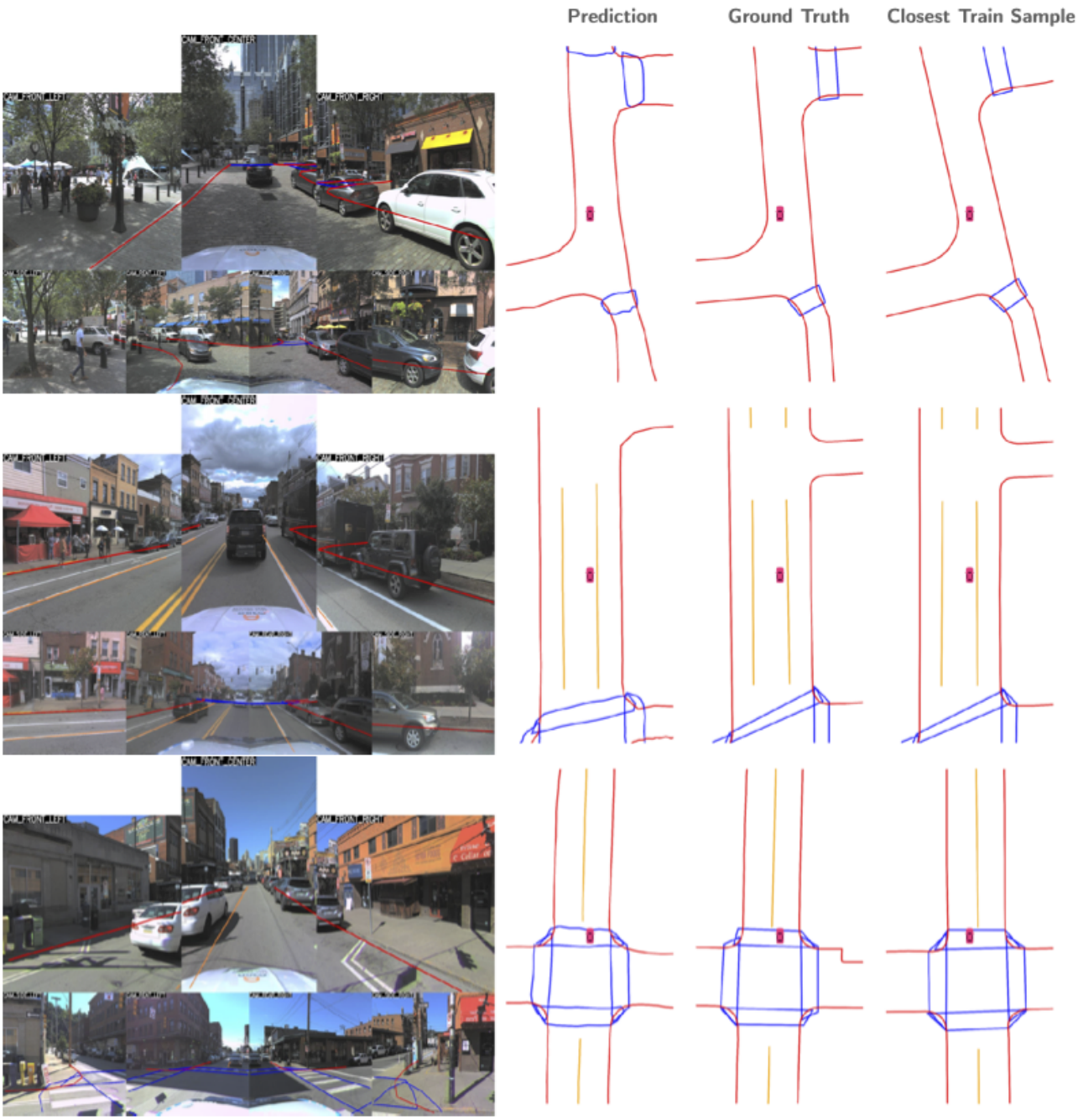


Figure 18. Multiple instances of validation or test predictions from MapTRv2 on Argoverse 2, alongside corresponding ground truth and the ground truth of the nearest training sample. The close similarities between the nearest training samples and the evaluation samples are apparent in each example. In the top illustration, the closest training sample exhibits a slight rotation, but the positions are very similar. In the bottom example, the closest training sample is from the lane adjacent to the evaluation sample

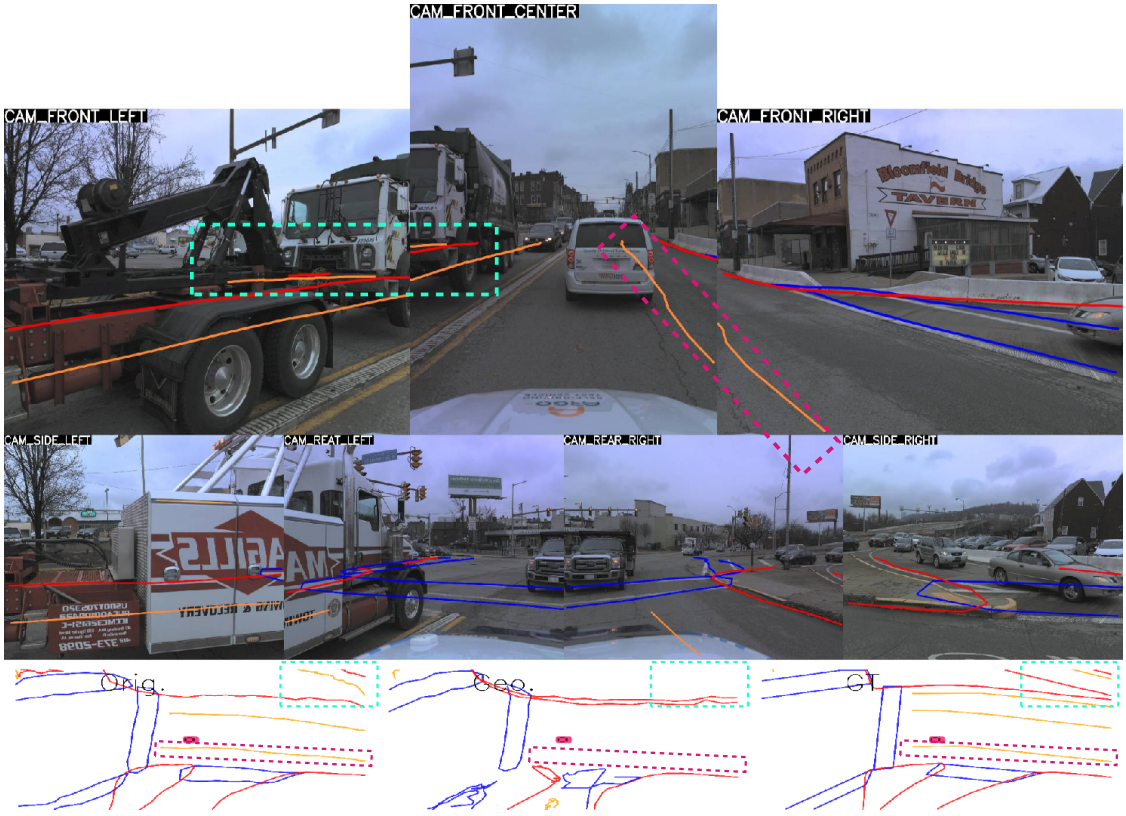


Figure 19. Argoverse 2 test prediction by mapTRv2 trained on Original (Orig.) and Geographical (Geo.), here Near Extrapolation, splits along with the ground truth (GT). Dividers, Boundaries, and Pedestrian crossings are visualized in orange, red, and blue, respectively. The predictions in the image view are from training on the Original split. Here, the predictions behind the truck on the left side, most notably the divider and boundary highlighted with the teal box, ought to be difficult to predict. Additionally, the model effectively predicts the lane divider to the right of the ego vehicle, highlighted by the pink box, even though there is no visible lane divider present in the image. It is worth noting that this may not solely be due to memorization, as the model could learn, *e.g.*, consistent data annotations and hints from road dividers and road width to accurately predict this non-visible lane divider.