SemVecNet: Generalizable Vector Map Generation for Arbitrary Sensor Configurations

Narayanan Elavathur Ranganatha*[1], Hengyuan Zhang*[1], Shashank Venkatramani[1], Jing-Yan Liao[1] and Henrik I. Christensen[2]

***Abstract*— Vector maps are essential in autonomous driving for tasks like localization and planning, yet their creation and maintenance are notably costly. While recent advances in online vector map generation for autonomous vehicles are promising, current models lack adaptability to different sensor configurations. They tend to overfit to specific sensor poses, leading to decreased performance and higher retraining costs. This limitation hampers their practical use in real-world applications. In response to this challenge, we propose a modular pipeline for vector map generation with improved generalization to sensor configurations. The pipeline leverages probabilistic semantic mapping to generate a bird's-eye-view (BEV) semantic map as an intermediate representation. This intermediate representation is then converted to a vector map using the MapTRv2 decoder. By adopting a BEV semantic map robust to different sensor configurations, our proposed approach significantly improves the generalization performance. We evaluate the model on datasets that are different from the training set including real-world data collected with our platform with different sensor configurations and show that the model generalizes significantly better than the state-of-the-art methods. The code will be available at https://github.com/AutonomousVehicleLaboratory/SemVecNet**

I. INTRODUCTION

Countless applications await the deployment of autonomous driving technology. For instance, addressing the truck driver shortages in the logistic sector, reducing the need for parking spaces and most importantly, improving driving safety. Current autonomous vehicle technology relies on high-definition (HD) maps. HD maps, as shown in Fig. 1, consist of accurately localized road features, including lane lines, crosswalks, sidewalks, and centerlines [1][2][3]. These highly detailed maps not only are essential for localization and planning, but also provide a rich context for detection, tracking, and prediction [4]. For example, a local planner follows the lane labels while avoiding collision with other road users and the crosswalk labels could serve as a strong prior for pedestrian interaction.

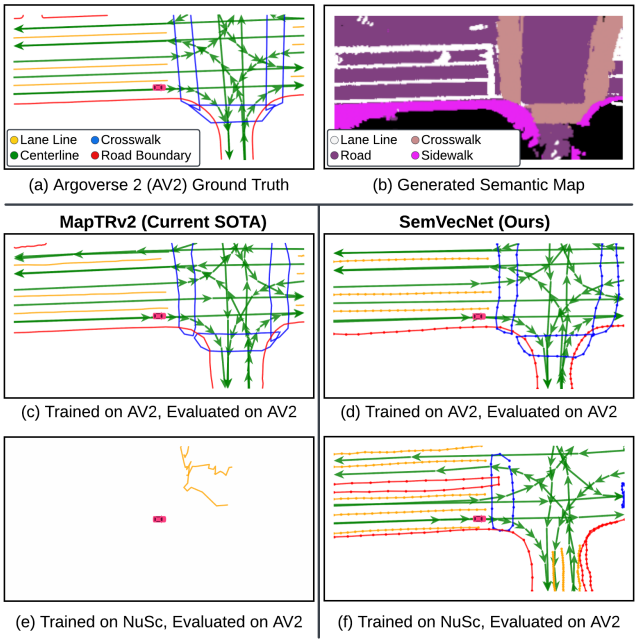


Fig. 1. State-of-the-art vector map generation models, for example, MapTRv2, perform well when trained and evaluated on the same dataset, but their performance degrades significantly when evaluated on a different dataset. Data labeling and retraining are required which limits their real-world application. SemVecNet shows significant improvements in performance transfer leveraging the semantic mapping with more robust sensor configuration generalization as an intermediate representation.

Traditionally, HD map creation involves substantial human annotation efforts; however, the dynamic nature of urban environments renders human involvement an ongoing expense. This predicament prompts two potential strategies: a transition to sparse-definition (SD) maps or the automation of HD map generation. While the former option may prove valuable in tasks like localization [5][6] and navigation [7][8], prior research [9][10] emphasizes the enduring significance of lane-level information, particularly in complex scenarios like intersections, for trajectory prediction. Some methods propose to automate the HD map generation offline [11]. However, with about 15,000 miles (24,000 kilometers) increase in urban roads or streets per year in the US from

*These authors contributed equally to this work.

**This research is supported by Nissan and Qualcomm.

[1]Students of Contextual Robotics Institute, University of California San Diego, La Jolla, CA 92093, USA {nelavathurranganatha, hyzhang, svenkatramani, j3liao}@ucsd.edu

[2]Henrik I. Christensen is with Faculty of the Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA 92093, USA hichristensen@ucsd.edu

2011-2021 [12], generating offline HD maps is not scalable.

In contrast, contemporary efforts [13][14][15][16][17] have pursued directly generating HD maps online, showcasing substantial efficacy within specific sensor configurations. Nevertheless, the primary challenge observed across these approaches is poor generalization to sensor configurations that are disparate from the data on which these approaches train. As shown in Fig. 1, when MapTRv2 [15] is evaluated on a dataset that is different from the training dataset, the model output degrades significantly. The poor generalization impedes the deployment of these models in real-world applications where changing of sensor locations, adding/removing sensors, and deploying on different platforms are common. These changes would incur costly data collection and retraining. Not to mention that this is all based on the assumption that HD map labels are available for retraining. For many including us, the cost is prohibitive.

We identify that the huge performance gap originates from the learned view transform modules [18][19]. These modules, despite trying to account for sensor configurations by taking camera parameters as input, inevitably overfit to the training set where the sensor configurations are often fixed, resulting in poor generalization on datasets with different sensor configurations.

Thus, rather than adopting an end-to-end approach with a view transform module, our work presents a modular pipeline designed to address the generalization issue. We leverage a semantic mapping approach that takes a 3D LiDAR point cloud, 2D images, and sensor intrinsics and extrinsics to generate intermediate semantic grid maps with improved generalization to different sensor configurations. The maps are further processed by a vectorization module into vector maps. This method, uses 3D geometry to mitigate overfitting and adapts to varying sensor setups, significantly improving generalization, as shown in Fig. 1. By standardizing input through this ego-centric BEV map, our model demonstrates enhanced adaptability in diverse real-world scenarios, particularly in vector map generation with unseen sensor platforms, without the need for model retraining.

The paper is organized with an initial discussion of related work in Section II. Then we present the overall pipeline in Section III, followed by the associated experiments and ablation studies in Section IV. Lastly, we summarize in Section V. Our key contributions are summarized as follows:

- **Generalized Sensor Configuration Vector Map Generation Pipeline:**
  Rather than projecting images into BEV through view transform networks, our approach leverages a BEV semantic map as an intermediate representation to refrain our pipeline from overfitting to any specific sensor configuration. This allows our proposed vector map generation pipeline to generalize to unseen sensor platforms.
- **Validation of Generalization Capability through Cross-Dataset and Real-world Experiments:**
  We test our pipeline on datasets with different sensor configurations including real-world data collected on our vehicle platform, which demonstrate that our proposed approach improves performance transfer significantly.

II. Related Work

The important role of maps in current autonomous driving architectures motivates extensive research in this area. Some studies focus on understanding scene semantics, which is a crucial step in building maps. Others attempt to directly generate maps with various levels of detail.

A. Semantic Mapping

Prior work explores building a semantic representation of the environment, derived either from a single frame or multiple frames. Single-frame approaches are especially challenging due to the occlusion by other road users and building structures. BEVFormer [19] addresses this issue by fusing surround-view-camera image features to create a consistent BEV feature for decoding objects and semantic maps. BEVFusion [20] further enhances this approach by integrating LiDAR point cloud data with image features. These methods demonstrate impressive performance in various downstream tasks, such as detection and mapping. However, their limitation arises from the view transform network overfitting to the training data, making model retraining a necessity when applied to a new setting with different sensor configuration settings, such as a different car with cameras at varied heights and viewing angles.

On the other hand, multi-frame approaches can leverage information extracted over time. Many early works focus on drivable regions [21][22] and often rely on planar assumptions [21]. More recently Paz et al. [23] proposed the fusion of point cloud maps and image semantic segmentations to generate a probabilistic semantic grid map. The pipeline is more generalizable to sensor configuration as it takes the camera intrinsics and extrinsics into account in the mapping process. In this study, we adopt the semantic maps as an intermediate representation. We extend the pipeline to utilize real-time point cloud and semantic segmentation to enable online mapping.

While semantic grid maps are informative, they require additional processing to be used in downstream tasks such as route planning and lane following. Additionally, they are less compact compared to vectorized maps, leading to storage and bandwidth overhead.

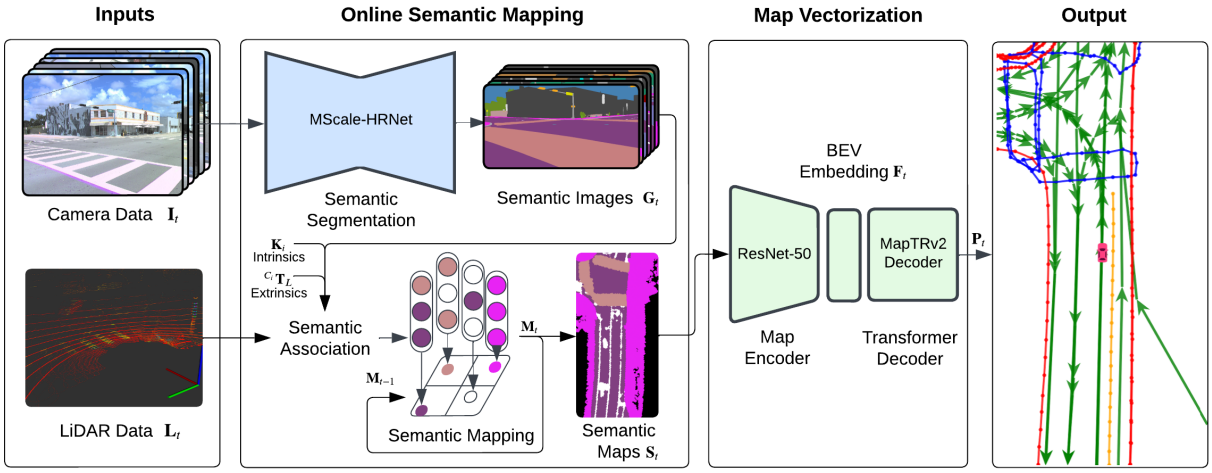


Fig. 2. SemVecNet takes camera images and LiDAR point cloud to generate a generalized sensor configuration BEV semantic map as an intermediate representation. The semantic map is vectorized into map elements such as centerlines, lane boundaries, crosswalks and road boundaries.

B. Online Vector Mapping

Vector maps encode map elements as polylines. There are online approaches and offline approaches. Offline approaches allow the aggregation of more information. For example, Zhou et al. [11] automate the HD building pipeline with instance segmentation, mapping, and particle filter based lane aggregation. However, they require ongoing maintenance.

Recent methods explore online HD map construction to avoid the need for continuous map maintenance. Li et al. propose HDMapNet [13], which generates a semantic representation with instance features, and then post-processes it to generate vectorized maps. Liu et al. propose VectorMapNet [24] to decode vector representation directly without the intermediate semantic representation. Liao et al. propose MapTR [14] and MapTRv2 [15] which use permutation invariance loss to supervise the vectorized map element generation process.

A new trend in online mapping not only estimates the map elements but also detects traffic elements and understands relations between them [2][16][17]. For example, in the OpenLaneV2 [2] dataset, traffic signs in images are associated with the centerlines of the lanes under their control. TopoNet [16] uses a scene graph neural network to model the association.

In our research, our focus is solely on the map element estimation task. While existing architectures achieves great performance when trained and tested on the same dataset, they often fall short in generalizing to other sensor settings. In contrast, our approach bridges the gap by leveraging a semantic mapping pipeline more robust to different sensor configurations to generate an intermediate semantic representation, and the final map is subsequently derived by our vectorization pipeline.

III. Approach

We hypothesize the root cause that prohibits existing work from generalizing across different sensor setups is transformation overfitting. As the network is only trained with one specific sensor configuration in the dataset, they tend to overfit to the coordinate transformation between the sensors and the world. To address the generalization issue in vector map creation approaches, we propose to combine semantic mapping that considers varying intrinsic and extrinsic configurations to generate an intermediate semantic map, and then decode the semantic map into the vector representations.

Formally, the problem is defined as follows: given a set of n images $\mathbf{I}_t = \{I_1, I_2, \ldots, I_n\}$, LiDAR point clouds $\mathbf{L}_t$ and semantic map prior $\mathbf{M}_{t-1}$ at timestep t, generate a collection of m point sets $\mathbf{P}_t = \{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_m\}$ to represent road elements such as lane-markings, pedestrian crossing, and road edges. The initial semantic map prior $\mathbf{M}_0$ is all zeros and in each timestamp the semantic map is also updated.

We describe the two major components of SemVecNet, semantic mapping and map vectorization in Sub-section III-A and Sub-section III-B.

A. Real-time Semantic Mapping

We introduce the generalized sensor configuration semantic mapping to the online vector mapping pipeline. The semantic mapping pipeline generates ego-centric Birds-Eye-View (BEV) semantic map $\mathbf{S}_t \in \mathbb{R}^{H_{bev} \times W_{bev} \times 3}$, where the color channel represents the map types such as road, lane markings, crosswalks and sidewalks. Building on prior work [23][25], we further make the pipeline real-time and online. It comprises of three key components: semantic segmentation, semantic association, and semantic mapping.

Semantic Segmentation: The semantic segmentation model takes images $\mathbf{I}_t$ and generates semantic masks $\mathbf{G}_t$. We

use HRNet+OCR (MScale-HRNet) [26] as the model due to its highly accurate segmentations, yielding cleaner semantic maps. We leverage TensorRT [27] to optimize MScale-HRNet, which reduce the inference time significantly without noticeable performance degradation. This optimization results in the mapping pipeline operating at a frequency exceeding 10 Hz without compromising data quality.

Semantic Association: The semantic association module projects LiDAR point cloud $\mathbf{L}_t$ onto the semantic images $\mathbf{G}_t$ to associate depth with semantic, resulting in a semantic point cloud with accurate geometry. Given the camera intrinsics $\mathbf{K}_i$ and extrinsics ${}^{C_i}\mathbf{T}_L$ for the camera i, the projection of a point $\mathbf{x}_L$ in LiDAR into a pixel in image $\mathbf{x}_I$ is given by

$$\mathbf{x}_I = \mathbf{K}_i[\mathbf{I}_3|\mathbf{0}]^{C_i}\mathbf{T}_L\mathbf{x}_L, \tag{1}$$

where the extrinsics ${}^{C_i}\mathbf{T}_L$ is a transformation from LiDAR frame to Camera frame. Different from prior approach [23][25] where they use LiDAR point map, we use real-time point cloud which eliminates the point cloud map building process, enabling it to be used in online mapping.

Probabilistic Mapping: The probabilistic mapping module takes the semantic point cloud $\mathbf{C}_t$ and integrates it into a BEV probabilistic grid map $\mathbf{M}_t \in \mathbb{R}^{H_{bev}\times W_{bev}\times C_{bev}}$, where C_{bev} is the number of class labels. The grid map is then rendered into a semantic map $\mathbf{S}_t$ with the highest probabiltiy class.

$$P(c_t|\mathbf{M}_{t-1}, z_t, i_t) = \frac{1}{N_m}P(z_t|c_t)P(i_t|c_t)P(c_{t-1}|\mathbf{M}_{t-1}), \tag{2}$$

with z_t represents the observed semantic label for the grid at time t, i_t represents the observed intensity and N_m represents the normalization factor. The confusion matrix of the model $P(z_t|c_t)$ and LiDAR intensity prior $P(i_t|c_t)$ are accounted for in the process.

Combined with suitable SLAM methods [28][29], the semantic mapping pipeline can run online at more than 10Hz and build a semantic map with a single camera and LiDAR. Our approach leverages projective geometry and therefore can build generalized sensor configuration semantic representation with different cameras and LiDARs. The semantic map with roads, lanes, crosswalks and sidewalks can then be used for vector mapping.

B. Map Vectorization

Given the generated BEV semantic map $\mathbf{S}_t \in \mathbb{R}^{H_{bev}\times W_{bev}\times 3}$, the Map Vectorization Model (MVM) takes $\mathbf{S}_t$ and maps it to vector map elements $\mathbf{P}_t = [\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_m]$. Each vector map element is modeled as a point set $\mathbf{P}_j = [p_0, p_1, \ldots, p_n]$. We utilize the equivariant permutations as stated in [15] to deal with the ambiguity of multiple correct permutations for a $\mathbf{P}_j$.

The MVM has an encoder-decoder format. Given $\mathbf{S}_t$, a feature map $\mathbf{F}_t \in \mathbb{R}^{H_c\times W_c\times C}$ is generated using a encoder. This encoder in principle can be any model that takes an image as input and outputs a feature representation. We adopt the Resnet-50 [30] model as our image encoder with reduced strides to maintain the dimension. $\mathbf{F}_t$ acts as BEV embedding which is then passed into the decoder.

The decoder following a similar structure to the one in MapTRv2 [15], consists of a transformer decoder that uses map queries. These queries consist of instance-level queries $Q_i = \{q_i^j\}_{j=1}^m$ as well as point-level queries $Q_p = \{q_p^j\}_{j=1}^n$ that each instance-level query uses. These are then summed to form hierarchical queries, so for map element k, we get $Q_h^k = \{q_h^j\}_{j=1}^n = \{q_i^k + q_p^j\}_{j=1}^n$. The decoder uses self-attention within the map queries to facilitate information exchange between instance-level queries and also the point-level queries within the instances. The decoder then uses cross-attention to facilitate the interaction between the map queries and $\mathbf{F}_t$.

The output of the decoder is then passed into a classification branch which outputs the instance scores and a point regression branch which produces the 2D coordinates of the n points in the point set. This way each query Q_h^k is mapped to a class c_{vk} = { Pedestrian Crossing, Lane Divider, Lane Boundary, Centerline} and each hierarchical query Q_h^{kj} is mapped to a 2D BEV point $p_{kj} = \{x_{kj}, y_{kj}\}$.

Loss: To supervise the MVM, following MapTRv2 [15], we perform instance-level matching and then point-level matching. Let $\hat{\sigma}_k$ denote the optimal matching and ordering for the k^{th} map element in the ground truth. Therefore $\hat{c}_{v\hat{\sigma}_k}$ denotes optimal instance matching to ground truth element k and $\hat{P}_{\hat{\sigma}_k} = \{\hat{p}^j_{\hat{\sigma}_k}\}_{j=1}^m$ represents optimal ordering of the points once instance matching is done. $\hat{e}^j_{\hat{\sigma}_k}$ denotes the predicted edge between the points $\hat{p}^j_{\hat{\sigma}_k}$ and $\hat{p}^{(j+1)\,mod\,n}_{\hat{\sigma}_k}$ and e^j_k denotes the predicted edge between the points p^j_k and $p^{(j+1)\,mod\,n}_k$. We follow [15] and use the focal loss l_{focal_loss} [31] for the instance label classification l_{cls}, point-to-point loss l_{p^2p} for each predicted point and an edge direction loss l_{dir} to supervise the direction of the edge connecting two points. We also use the auxillary BEV segmentation loss l_{Seg} introduced in MapTRv2. For this a auxillary segmentation head is added which is denoted as $\mathbf{s}(\cdot)$. We do not use the auxillary perspective view segmentation depth estimation losses mentioned in MapTRv2 as the perspective view is not part of the input. The losses are defined as follows:

$$l_{cls} = \sum_{k=1}^{m} l_{focal\_loss}(\hat{c}_{v\hat{\sigma}_k}, c_{vk}) \tag{3}$$

$$l_{p^2p} = \sum_{k=1}^{m} \mathbf{1}_{c_k\neq\phi} \sum_{j=1}^{n} D(\hat{p}^j_{\hat{\sigma}_k}, p^j_k) \tag{4}$$

$$l_{dir} = \sum_{k=1}^{m} \mathbf{1}_{c_k\neq\phi} \sum_{j=1}^{n} cos(\hat{e}^j_k, e^j_k) \tag{5}$$

$$l_{Seg} = l_{CE}(\mathbf{s}(\mathbf{F}_t), \mathbf{GT}_{BEV}) \tag{6}$$

where $D()$ denotes the Manhattan distance and $cos()$ denotes cosine similarity. l_{CE} is the cross-entropy loss and $\mathbf{GT}_{BEV}$ denotes the ground truth BEV Segmentation. Therefore the final loss l becomes:

$$l = w_{cls}l_{cls} + w_{p^2p}l_{p^2p} + w_{dir}l_{dir} + w_{Seg}l_{Seg} \tag{7}$$

IV. Experiments

In this section, we perform experiments to validate the effectiveness of SemVecNet. We introduce the datasets, evaluation metrics and training configurations in Sub-section IV-A. Then we perform experiments to show the generalization performance with cross-dataset experiments in Sub-section IV-B and real-world data in Sub-section IV-C. Additionally we present various ablation studies to show the key factors that affect the design in Sub-section IV-D. Lastly we discuss limitations and potential improvements in Sub-section IV-E.

A. Datasets, Metric and Training

Datasets: We conducted evaluations of our method using two prominent datasets: Argoverse2 [3] and the NuScenes [1] dataset. Argoverse2 offers 3D Vector Maps for each 15-second log, featuring images sampled at 20Hz and LiDAR data sampled at 10Hz. The dataset comprises 700 training logs, 150 validation logs, and 150 test logs. In our approach with Argoverse2, we operate at the frequency of the LiDAR sensor, utilizing data sampled at 10Hz and also only do 2D Vector Map estimation. On the other hand, NuScenes provides a collection of 1000 scenes, accompanied by city-level Vector Maps. In the case of NuScenes, we specifically leverage the samples dataset as it contains synchronised data, operating on sensor data sampled at 2Hz to train our model.

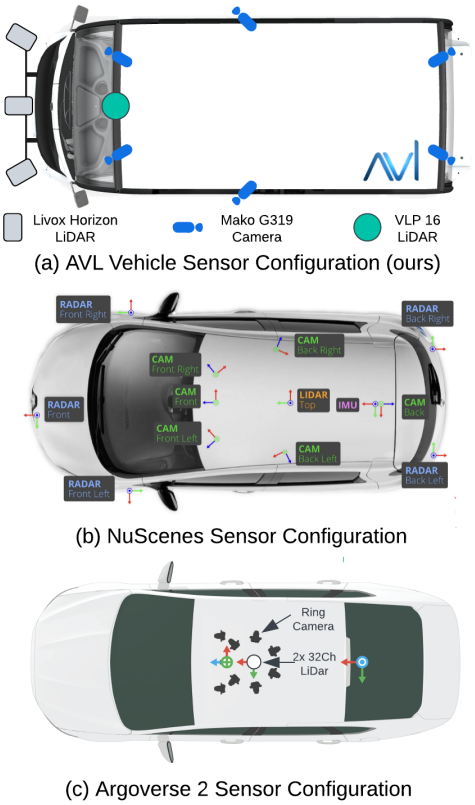
Metrics: We follow MapTRv2 [15] and use Average Precision (AP) metric for evaluation. The AP is calculated for *Pedestrian Crossing*, *Lane Divider*, *Road Boundary* and *Centerlines* separately and averaged. The AP is calculated with chamfer distance $D_{Chamfer}$ across three thresholds $\theta \in \{0.5m, 1.0m, 1.5m\}$. The Chamfer distance is given by

$$D_{Chamfer}(X,Y) = \frac{1}{|X|}\sum_{x\in X}\min_{y\in Y}||x-y||_2^2 + \frac{1}{|Y|}\sum_{y\in Y}\min_{x\in X}||y-x||_2^2. \tag{8}$$

The AP for each class is averaged to give the Mean Average Precision (mAP), given by

$$AP = \frac{1}{|\theta|}\sum_{\lambda\in\theta} AP_\lambda. \tag{9}$$

Training: For the Semantic Segmentation Network, we use HRNet-OCR trained on the Mapilary dataset [32] as described in [25]. For the decoder, we remove the perspective view to BEV view transfomation from the baseline MapTRv2 model and pass encoded BEV features to the MapTRv2 Decoder. We follow the same learning rate scheduler as MapTRv2 but utilize cyclic momentum as we observed that



(a) AVL Vehicle Sensor Configuration (ours)

(b) NuScenes Sensor Configuration

(c) Argoverse 2 Sensor Configuration

Fig. 3. The diagrams (a), (b), and (c) display the significant configuration changes across platforms in sensor poses and number of sensors, for both LiDAR and cameras in NuScenes [1], Argoverse 2 [3] and AVL.

led to better convergence. Loss weghts are $w_{cls} = 2.0, w_{p2p} = 5.0, w_{dir} = 0.005, w_{Seg} = 1.0$ following [15]. The model takes 12 hours to converge on 8 NVIDIA A-10 GPUs for both datasets with a batch size of 5 on each GPU.

B. Cross-dataset Experiments

We are particularly interested in quantitatively measuring the model's performance generalization capabilities. This is one of the most critical issues faced in real-world applications. A model with great performance only on its trained sensor configuration has a prohibitive cost to adapt to new platforms and will not be scalable in the long run.

To this end, we conduct cross-dataset experiments and use the mAP ratio to measure the performance transfer. In cross-dataset experiments, a model trained on dataset A is evaluated on the validation set of dataset A and B. The ratio of the mAP on the test set of B to A measures performance transfer.

This task is especially challenging since the cross-dataset sensor configurations are often drastically different. For example, NuScenes dataset has six cameras and a LiDAR and Argoverse 2 has seven cameras and two LiDARs, as shown in Fig. 3. These sensors are also placed differently, resulting in varying viewing angles. This explains why MapTRv2, the

current SOTA for vector map generation, achieves an mAP ratio of 0, as shown in Table I. Comparatively, SemVecNet obtains a performance transfer of 24.8% when evaluating an AV2 trained model on NuScenes, and a 33.1% performance transfer when evaluating a NuScenes model on AV2.

From our cross-dataset results, it is apparent that the standard MapTRv2 formulation overfits to the camera configuration of the training setup, limited by learned view transform modules. Alternatively, SemVecNet has a more standardized intermediate representation of semantic maps across datasets, improving performance transfer.

TABLE I

CROSS-DATASET PERFORMANCE TRANSFER

Model	Train	Test	mAP ↑	mAP ratio (%) ↑
SemVecNet	AV2	NuSc	12.2	**24.8**
	AV2	AV2	49.0	
MapTRv2	AV2	NuSc	0	0
	AV2	AV2	**67.4**	
SemVecNet	NuSc	AV2	16.2	**33.1**
	NuSc	NuSc	48.8	
MapTRv2	NuSc	AV2	0	0
	NuSc	NuSc	**61.5**	

C. Real-World Experiments

Aside from demonstrating our systems generalization capabilities between datasets, we also conducted experiments with data we collected on UC San Diego campus. The sensor configuration is shown in Fig. 3, and other details of our autonomous driving platform can be found in [33].

Given that it is costly to generate groundtruth HD map labels, we only show qualitative results on the campus data to complement our cross-dataset quantitative evaluation. From Fig. 4, we can see that our pipeline is still able to generate meaningful output without fine-tuning the model with our data. The centerlines and road boundaries are captured accurately for the most part. This experiment shows that our pipeline can work with sensor configurations that are vastly different from the training domain.

D. Ablation Studies

We perform experiments to validate various design choices such as single view vs surround view and map resolution. These experiments are performed on Argoverse 2 dataset [3].

Single View vs Surround View: Our pipeline can adapt to single-camera or multi-camera input. For single-camera input, we take the front camera images and crop them to maintain a similar aspect ratio as a typical semantic segmentation input. For surround-view input, the semantic point cloud incorporates semantic labels from all views to generate the probabilistic semantic map.

Compared to maps generated from single-view images, the maps generated from surround-view images have more coverage. Better coverage provides more context, especially for the initial few frames in each scene, as shown in Fig. 5. The lower coverage causes a performance decrease of 28%.

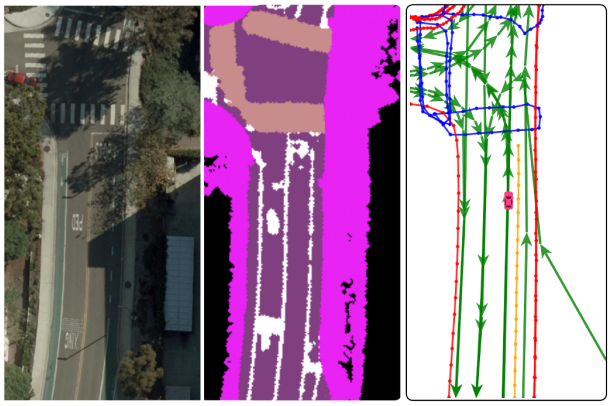


Fig. 4. The qualitative result by directly inference from campus data. The images from top to bottom are satellite image [34], BEV semantic map, and vector map output from SemVecNet from the same region on UC San Diego campus.

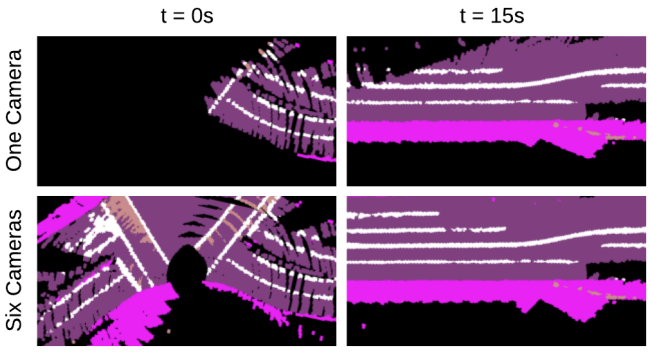


Fig. 5. Top row represents semantic map made with a single camera. The bottom row represents semantic maps made with six cameras. At the start of a log (first column), a lot of information is lost if some cameras are left out. The map by the end of the log (second column) ends up looking similar.

TABLE II

EFFECT OF NUMBER OF CAMERAS

	Number of Cameras	mAP ↑
Argoverse 2	1	35.2
	6	49.0

We find it interesting that our approach can maintain this level of performance despite reducing the camera number to 1. The temporal aggregation baked into the semantic mapping pipeline allows us to maintain a large portion of our performance even if we reduce the number of cameras.

Effect of Semantic Map Resolution: As mentioned in III-A, grid resolution for the semantic map is a hyhperparameter that influences this map creation process. In [23] the grid

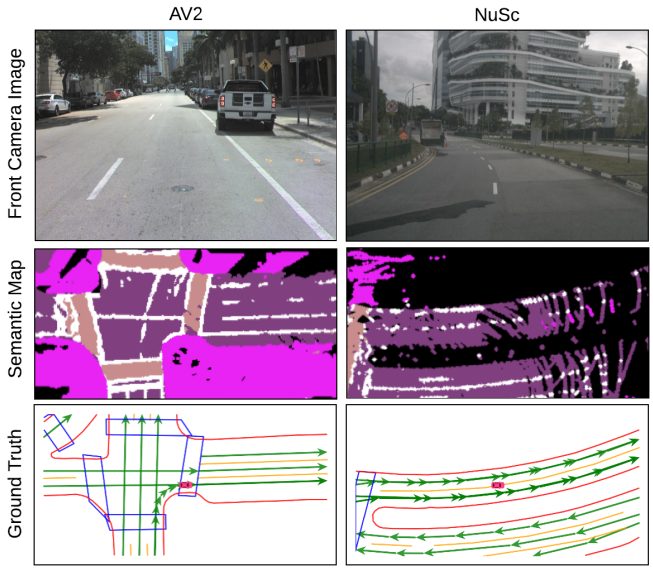


Fig. 6. This figure shows a significant distribution shift between AV2 and NuScenes. NuScenes semantic maps are sparser due to sparser LiDAR observations. Additionally, NuScenes includes left-hand drive in Singapore logs, while Argoverse 2 is solely U.S. right-hand drive.

pixel size is 0.2 meters, however we observe feature blurring in the semantic maps. To try and mitigate this we reduced the pixel size to 0.1 meters. As shown in Table III, the higher grid resolution doesn't improve performance while significantly increasing the computation. The lack of performance improvement could be due to reduced probabilistic correction, or the network not benefiting sharper images. Based on the results, we use 0.2 resolution for other experiments.

	Pixel Size	mAP ↑
Argoverse 2	0.1	48.9
	0.2	49.0

TABLE III
RESOLUTION PERFORMANCE

E. Discussion and Limitations

Our proposed approach addresses the critical issue of designing online vector mapping pipelines that are more generalizable improving cross-dataset performance significantly, thus enabling the model to be applied to a new domain without costly labeling and retraining. While our model is designed to be more robust to different camera intrinsics and extrinsics, the performance is still affected by variations in data distribution.

The first variation is the density of LiDAR observations. AV2 has two stacked 32 channel LiDARs (64 channels effective) while NuScenes has a single 32 channel LiDAR. This causes sparser semantic maps for NuScenes, as seen in Fig. 6. These semantic map holes presents a domain gap for inputs to the Vector Map generation, and likely hinders cross dataset performance.

Additionally, AV2 is recorded in U.S. cities while NuScenes is recorded both in the U.S. and Singapore. This presents a secondary domain gap in road network distribution. For example, AV2 typically includes a crosswalk at nearly every intersection, a pattern not consistently observed in NuScenes. Consequently, the model trained on AV2 data tends to over-predict the presence of pedestrian crossings at intersections when applied to NuScenes data, leading to reduced accuracy in identifying pedestrian crossings as seen in Table IV. The NuScenes logs from Singapore also include left-hand driving scenarios, which is fundamentally different from Argoverse 2's right-hand driving, leading to opposite centerline directions in cross-dataset visualizations.

Train	Test	AP ↑				mAP ↑
		ped.	div.	bou.	cent.	
AV2	NuSc	5.0	8.7	15.7	19.4	12.2
AV2	AV2	40.7	51.3	53.4	50.6	49.0
NuSc	AV2	3.5	13.1	27.8	20.2	16.2
NuSc	NuSc	41.4	50.5	54.5	49.0	48.8

TABLE IV
INDIVIDUAL CLASS PERFORMANCE TRANSFER

We believe the sparse semantic maps and presence of more diverse road structures in NuScenes likely leads to better generalization and higher performance of NuScenes models on AV2, than AV2 models on NuScenes.

However, our model shows inferior performance when evaluated on the same dataset. This can be caused by the information loss introduced in the intermediate map representation. Future directions involve using probabilistic grids or neural features as the intermediate representation to reduce information loss and improve performance.

V. CONCLUSIONS

We introduce SemVecNet, an online vector map generation pipeline designed to be more robust to sensor configurations. Through cross-dataset evaluations, we show that SemVecNet significantly improved the generalization capability, supported by both quantitative and qualitative assessments. Furthermore, SemVecNet showcases its transferability through successful testing on real-world data collected from the UC San Diego campus. We believe this is a step towards sensor-configuration-agnostic autonomous system design that is more scalable, enabling algorithms to be deployed in systems with various sensor configurations.Future work involves characterizing the generalization problem arising from view-transform modules in other tasks such as detection and semantic segmentation, and reduce the gap in the proposed pipeline.

REFERENCES

[1] 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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, Seattle, WA, USA, 13-19 June 2020.

[2] Huijie Wang, Tianyu Li, Yang Li, Li Chen, Chonghao Sima, Zhenbo Liu, Yuting Wang, Shengyin Jiang, Peijin Jia, Bangjun Wang, Feng Wen, Hang Xu, Ping Luo, Junchi Yan, Wei Zhang, and Hongyang Li. Openlane-v2: A topology reasoning benchmark for unified 3d hd mapping. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 18873–18884. Curran Associates, Inc., 2023.

[3] 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 *Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, online, 06 – 14 December 2021.

[4] Vidyaa Krishnan Nivash and Ahmed H. Qureshi. Simmf: Semantics-aware interactive multiagent motion forecasting for autonomous vehicle driving. *arXiv preprint arXiv:2306.14941*, 2023.

[5] Younghun Cho, Giseop Kim, Sangmin Lee, and Jee-Hwan Ryu. Openstreetmap-based lidar global localization in urban environment without a prior lidar map. *IEEE Robotics and Automation Letters*, 7(2):4999–5006, 2022.

[6] Philipp Ruchti, Bastian Steder, Michael Ruhnke, and Wolfram Burgard. Localization on openstreetmap data using a 3d laser scanner. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 5260–5265. IEEE, 2015.

[7] Matthias Hentschel and Bernardo Wagner. Autonomous robot navigation based on openstreetmap geodata. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1645–1650. IEEE, 2010.

[8] KAFA Samah, S Ibrahim, N Ghazali, M Suffian, M Mansor, and WA Latif. Mapping a hospital using openstreetmap and graphhopper: A navigation system. *Bulletin of Electrical Engineering and Informatics*, 9(2):661–668, 2020.

[9] Julian Schmidt, Julian Jordan, Franz Gritschneder, Thomas Monninger, and Klaus Dietmayer. Exploring navigation maps for learning-based motion prediction. *arXiv preprint arXiv:2302.06195*, 2023.

[10] Jing-Yan Liao, Parth Doshi, Zihan Zhang, David Paz, and Henrik Christensen. Osm vs hd maps: Map representations for trajectory prediction. *arXiv preprint arXiv:2311.02305*, 2023.

[11] Yiyang Zhou, Yuichi Takeda, Masayoshi Tomizuka, and Wei Zhan. Automatic construction of lane-level hd maps for urban scenes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6649–6656, Prague, Czech Republic, 27 September - 01 October 2021.

[12] Federal Highway Administration U.S. Department of Transportation. Highway statistics. `http://www.fhwa.dot.gov/policyinformation/statistics.cfm`, last accessed on Feb 01, 2024.

[13] 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, Philadelphia, PA, USA, 23-27 May 2022.

[14] 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 *International Conference on Learning Representations*, 2023.

[15] 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. *arXiv preprint arXiv:2308.05736*, 2023.

[16] Tianyu Li, Li Chen, Huijie Wang, Yang Li, Jiazhi Yang, Xiangwei Geng, Shengyin Jiang, Yuting Wang, Hang Xu, Chunjing Xu, Junchi Yan, Ping Luo, and Hongyang Li. Graph-based topology reasoning for driving scenes. *arXiv preprint arXiv:2304.05277*, 2023.

[17] Dongming Wu, Jiahao Chang, Fan Jia, Yingfei Liu, Tiancai Wang, and Jianbing Shen. Topomlp: A simple yet strong pipeline for driving topology reasoning. *arXiv preprint arXiv:2310.06753*, 2023.

[18] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. *arXiv preprint arXiv:2008.05711*, 2020.

[19] 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. *arXiv preprint arXiv:2203.17270*, 2022.

[20] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, ExCeL, London, UK, 29 May - 2 June 2023.

[21] S. Sengupta, P. Sturgess, L. Ladický, and P. H. S. Torr. Automatic dense visual semantic mapping from street-level imagery. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 857–862, Vilamoura-Algarve, Portugal, 07-12 Oct 2012.

[22] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. S. Torr. Urban 3d semantic modelling using stereo vision. In *2013 IEEE International Conference on Robotics and Automation*, pages 580–585, Karlsruhe, Germany, 06-10 May 2013.

[23] David Paz, Hengyuan Zhang, Qinru Li, Hao Xiang, and Henrik I. Christensen. Probabilistic semantic mapping for urban autonomous driving applications. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2059–2064, Las Vegas, NV, USA, 24 October 2020.

[24] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. *arXiv preprint arXiv:2206.08920*, 2023.

[25] Hengyuan Zhang, Shashank Venkatramani, David Paz, Qinru Li, Hao Xiang, and Henrik I. Christensen. Probabilistic semantic mapping for autonomous driving in urban environments. *Sensors*, 23(14), 2023.

[26] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation, 2020.

[27] NVIDIA. Tensorrt open source software. `https://github.com/NVIDIA/TensorRT`, last accessed on Feb 01, 2024.

[28] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans. Robot.*, 37(6):1874–1890, 2021.

[29] Kailai Li, Meng Li, and Uwe D. Hanebeck. Towards high-performance solid-state-lidar-inertial odometry and mapping. *IEEE Robotics and Automation Letters*, 6(3):5167–5174, 2021.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, 27-30 June 2016. IEEE.

[31] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020.

[32] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5000–5009, Venice, Italy, 22-29 October 2017.

[33] Henrik Christensen, David Paz, Hengyuan Zhang, Dominique Meyer, Hao Xiang, Yunhai Han, Yuhan Liu, Andrew Liang, Zheng Zhong, and Shiqi Tang. Autonomous vehicles for micro-mobility. *Auton. Intell. Syst.*, 1(11):1–35, Nov 2021.

[34] Ersi. World imagery. `https://www.arcgis.com/apps/mapviewer/index.html?layers=10df2279f9684e4a9f6a7f08febac2a9`, last accessed on Feb 01, 2024.