Accelerating Online Mapping and Behavior Prediction via Direct BEV Feature Attention

Xunjiang Gu[1], Guanyu Song[1], Igor Gilitschenski[1,2], Marco Pavone[3,4], and Boris Ivanovic[3]

[1]University of Toronto [2]Vector Institute [3]NVIDIA Research [4]Stanford University

{alfred.gu, guanyu.song}@mail.utoronto.ca, gilitschenski@cs.toronto.edu, {mpavone, bivanovic}@nvidia.com, pavone@stanford.edu

Abstract. Understanding road geometry is a critical component of the autonomous vehicle (AV) stack. While high-definition (HD) maps can readily provide such information, they suffer from high labeling and maintenance costs. Accordingly, many recent works have proposed methods for estimating HD maps online from sensor data. The vast majority of recent approaches encode multi-camera observations into an intermediate representation, e.g., a bird's eye view (BEV) grid, and produce vector map elements via a decoder. While this architecture is performant, it decimates much of the information encoded in the intermediate representation, preventing downstream tasks (e.g., behavior prediction) from leveraging them. In this work, we propose exposing the rich internal features of online map estimation methods and show how they enable more tightly integrating online mapping with trajectory forecasting[1]. In doing so, we find that directly accessing internal BEV features yields up to **73%** faster inference speeds and up to **29%** more accurate predictions on the real-world nuScenes dataset.

Keywords: Autonomous Driving · Online HD Map Estimation · Behavior Prediction

1 Introduction

Perceiving the static environment surrounding an autonomous vehicle (AV) is a critical task for autonomous driving, providing geometric information (e.g., road layout) to downstream behavior prediction and motion planning modules. Traditionally, high-definition (HD) maps have served as the backbone for this understanding, offering centimeter-level geometries for road boundaries, lane dividers, lane centerlines, pedestrian crosswalks, traffic signs, road markings, and more. They have proven to be an indispensable part of enhancing AV situational awareness and navigational judgment in downstream prediction tasks. However, despite their undeniable utility, collecting and maintaining HD maps is labor-intensive and costly, which limits their scalability.

[1] Code: https://github.com/alfredgu001324/MapBEVPrediction

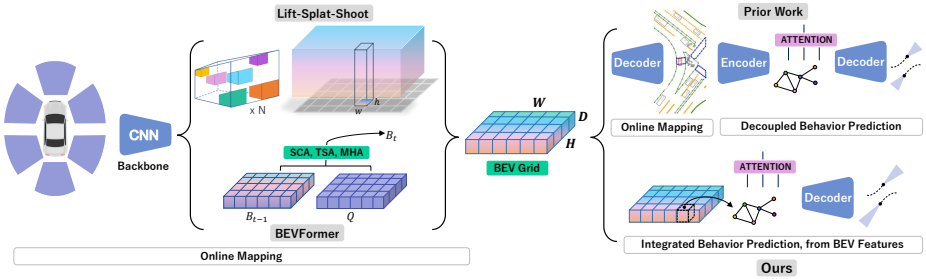

Fig. 1: Online map estimation approaches predominantly encode multi-camera observations into a canonical BEV feature grid prior to decoding vectorized map elements. In this work, we propose deeply integrating online mapping with downstream tasks through direct access to the rich BEV features of online map estimation methods.

In recent years, online HD map estimation methods have emerged as an alternative, aiming to predict HD map information directly from sensor observations. Starting from (multi-)camera images and optionally LiDAR pointclouds, state-of-the-art HD map estimation methods broadly employ an encoder-decoder neural network architecture (Fig. 1). An encoder first converts the sensor observations to a bird's eye view (BEV) grid of features. A decoder then predicts the location and semantic type of map elements from the BEV features. The resulting road geometries are commonly structured as combinations of polylines and polygons per map element type (e.g., road boundaries, lane dividers, pedestrian crosswalks). Such online-estimated maps serve as a practical substitute for offline HD mapping, providing necessary scene context for downstream tasks such as behavior prediction and motion planning. As an example, recent work [13] has shown success in coupling various map estimation methods with existing prediction frameworks, highlighting their potential to expedite the development of end-to-end AV stacks.

While such encoder-decoder approaches produce accurate HD maps, as we will show in Sec. 4, the attention mechanisms employed in decoding are computationally expensive (occupying the majority of model runtime) and do not produce outputs with associated uncertainty, which limits the ability of downstream modules to account for uncertainty. Moreover, such an architecture prevents downstream tasks from leveraging the rich intermediate features generated in the encoder's perspective-view-to-bird's-eye-view (PV2BEV) transformation, decimating information that cannot be described by point sets.

Contributions. Towards this end, we introduce three novel scene encoding strategies that leverage internal BEV features to improve the performance and accelerate the runtime of combined online mapping and behavior prediction systems. By directly leveraging BEV features, our proposed methods provide tighter integrations between map estimation and behavior prediction frameworks, achieving up to **73%** faster system inference speeds and an up to **29%** increase in downstream prediction accuracy on the real-world nuScenes dataset.

2 Related Work

2.1 Online Map Estimation

Online map estimation methods focus on generating a representation of the static environment surrounding an autonomous vehicle from its sensor data. Initial approaches used 2D BEV rasterized semantic segmentations as world representations. These maps were produced by either transforming observations to 3D and collapsing along the Z-axis [26, 28] or by utilizing cross-attention in geometry-aware Transformer [34] models [2, 20].

Recently, there has been a growth in vectorized map estimation methods that extend traditional BEV rasterization approaches. These methods employ an encoder-decoder architecture which regresses and classifies map elements in the form of polylines, polygons, and other curve representations [29]. Initial methods such as SuperFusion [5] and HDMapNet [19] combined both LiDAR point clouds and RGB images into a common BEV feature frame, with a subsequent hand-crafted post-processing stage to generate polyline map elements. To eliminate this post-processsing step, VectorMapNet [24] and InstaGraM [32] propose end-to-end models for vectorized HD map estimation.

In parallel, HD map estimation has also been formulated as a point set prediction task in the MapTR line of work [22, 23] and its extensions [36], yielding significant advancements in map estimation performance. To enable online inference from streaming observations, StreamMapNet [37] introduces a memory buffer that incorporates temporal data from prior timesteps. As many of these methods are commonly employed today, in this work we show how BEV features from multiple diverse mapping approaches can be leveraged to improve integrated system performance.

2.2 Map-Informed Trajectory Prediction

Learning-based trajectory prediction approaches initially leveraged rasterized maps for semantic scene context [30]. The rasterized map is treated as a top-down image and encoded via a convolutional neural network (CNN), concatenated with other scene context information (e.g., agent state history), and passed through the rest of the model [9, 16, 27, 31, 38]. Recently, state-of-the-art trajectory prediction methods have increasingly shifted to directly encoding raw polyline information from vectorized HD maps, demonstrating significant improvements in prediction accuracy. Initial approaches [8, 10, 11, 21, 39] utilized graph neural networks (GNNs) to encode lane polylines as nodes and their interactions with agent trajectories as edges. Extending this insight, Transformer [34] architectures with attention over map and agent embeddings have been widely adopted by current state-of-art methods [4, 12, 25, 40].

One recent related work investigates the integration between different combinations of map estimation and trajectory prediction models [13]. In it, they propose exposing uncertainties from map element regression and classification to downstream behavior prediction. In contrast to [13], our work focuses on

exposing information from an earlier stage of online mapping (immediately following observation encoding). As we will show in Sec. 4, our approach not only outperforms [13], it is also much more computationally efficient.

2.3 End-to-End Driving Architectures

End-to-end architectures present a promising approach for creating integrated stacks that can internally leverage more information, e.g., uncertainty, from upstream components. Recent works such as UniAD [15], VAD [18], and OccNet [33] demonstrate the feasibility and performance of incorporating both rasterized and vectorized HD map estimation within end-to-end driving. For example, UniAD [15] and OccNet [33] formulate online mapping as a dense prediction task, aiming to generate the semantics of map elements at a per-pixel or voxel granularity, whereas VAD focuses on producing vectorized HD map representations. In these architectures, the utility of mapping is twofold: it is both an auxiliary training task and an internal static world representation that aids downstream tasks. While these approaches lead to highly-integrated autonomy stacks, the use of rasterized or vectorized representations (rather than BEV features) leads to information loss and extra computational burden. Accordingly, our work is complementary in that our proposed strategies can be incorporated within end-to-end stacks to improve inference speeds as well as downstream prediction and planning accuracy.

3 Leveraging Online Mapping Features in Trajectory Prediction

As mentioned in Sec. 2, the majority of state-of-the-art online vectorized map estimation models adopt a BEV grid internally to featurize the surrounding environment in a geometry-preserving fashion. Our method focuses on leveraging these internal BEV representations by directly accessing them in trajectory prediction. In doing so, we improve the flow of information from mapping to prediction and can even accelerate the combined system's runtime by skipping map decoding altogether (depending on the predictor's need for lane information).

Encoding Observations: Feature extractors in map estimation models aim to transform inputs from various vehicle-mounted sensors (e.g., cameras and LiDAR) into a unified feature space. Note that our work focuses on multi-camera observations, in line with the majority of state-of-the-art map estimation approaches. Formally, given a set of multi-view images $I_t = \{I_1, \ldots, I_K\}$ at time t, map estimation models encode them using a standard backbone (e.g., ResNet-50 [14]) to generate corresponding multi-view feature maps $F_t = \{F_1, \ldots, F_K\}$. The 2D image features F_t are then converted into BEV features B_t using a PV2BEV transformation. The two most common PV2BEV approaches are based on BEVFormer [20] and Lift-Splat-Shoot (LSS) [28].

BEVFormer [20] is a Transformer-based architecture that converts multi-camera image features into BEV features. It employs a standard Transformer

encoder with specific enhancements: BEV queries $Q \in \mathbb{R}^{H \times W \times D}$, spatial cross-attention, and temporal self-attention. First, temporal information is queried from historical BEV features B_{t-1} through temporal self-attention,

$$\mathrm{TSA}(Q_p, \{Q, B_{t-1}\}) = \sum_{V \in \{Q, B_{t-1}\}} \mathrm{DeformAttn}(Q_p, p, V), \tag{1}$$

where $Q_p \in \mathbb{R}^D$ is the query for a single BEV grid point $p = (h, w)$. The queries Q are then employed to gather spatial information from the multi-camera features F_t via a spatial cross-attention mechanism,

$$\mathrm{SCA}(Q_p, F_t) = \frac{1}{|\mathcal{V}_{\mathrm{hit}}|} \sum_{i \in \mathcal{V}_{\mathrm{hit}}} \sum_{j=1}^{N_{\mathrm{ref}}} \mathrm{DeformAttn}(Q_p, \mathcal{P}(p, i, j), F_{i,t}), \tag{2}$$

where $\mathcal{V}_{\mathrm{hit}}$ denotes the camera views that contain p, $\mathcal{P}$ is the camera projection function from 3D world coordinates (h, w, and discrete height index j) to the 2D image plane of the i^{th} camera. This combined approach enables BEVFormer to efficiently understand temporal and spatial context, producing enhanced BEV features. As we will show in Sec. 4.2, incorporating temporal information in BEV features is quite beneficial for trajectory prediction.

Another common PV2BEV method is LSS [28]. Its first stage (Lift) featurizes individual images and converts them into a shared 3D frame via "unprojection", assigning multiple discrete depth points $(h, w, d) \in \mathbb{R}^3$ to each pixel in an image based on camera extrinsics and intrinsics. This forms a large point cloud with a 3D point at each depth per ray (HWD points). The second stage (Splat) aggregates these points into a common BEV feature grid using an efficient pillar pooling method.

Decoding Map Elements: To produce vectorized map elements, most map prediction models employ a Transformer-based decoder. They broadly consist of a hierarchical query embedding mechanism alongside Multihead Self Attention and Deformable Attention to accurately predict complex, irregular map elements from BEV features. Instance and point-level queries are combined for dynamic feature interaction, followed by classification and regression heads that predict the type and location of map element vertices, respectively. While such decoding architectures produce accurate maps, they are computationally expensive, and decoding occupies much of overall model runtime. MapTRv2 [23] attempts to address this by introducing more streamlined attention mechanisms in its decoder. In StreamMapNet [37], a Multi-Point Attention mechanism is utilized alongside a streaming approach that preserves previous queries and BEV features, aiming to improve map estimation performance by incorporating temporal information.

Behavior Prediction Models: Most state-of-the-art trajectory prediction models also leverage an encoder-decoder framework [30]. The encoder is responsible for capturing the scene's context, such as vectorized map elements (e.g., road edges and centerlines) as well as agent trajectories. The decoder then utilizes these encoded representations to forecast the future motion of agents in the scene. In the encoder, vectorized map elements are commonly encoded as either

nodes in a Graph Neural Network (GNN) or as tokens in a Transformer [34]. Two representative instantiations are DenseTNT [12] and HiVT [40], respectively.

At a high level, DenseTNT [12] leverages the VectorNet [8] hierarchical GNN context encoder to extract features from vectorized map elements. Agent trajectories and map element segments are first modeled as polyline subgraphs. Then, each resulting subgraph is further encoded as a node in a global GNN to capture their interactions. On the other hand, the Transformer-based HiVT [40] treats vectorized elements as sequences of tokens. Its hierarchical encoder consists of two stages: information within a local spatial window is encoded for each agent, followed by a global interaction encoder to model long-range interactions between agents.

Recent work [13] has explored strategies for coupling online-estimated vectorized maps and the above prediction models. However, as we will show in Sec. 4, their prediction performance and computational runtime can be further improved by harnessing the BEV features present within online map estimation models. Directly using BEV features provides prediction models access to richer information than the original decoded sets of polylines and polygons.

In the remainder of this section, we outline three different strategies for incorporating BEV features in downstream behavior prediction.

3.1 Modeling Agent-Lane Interactions via BEV Feature Attention

Inspired by the approach taken in Vision Transformer (ViT) [6], we treat map BEV features as an image, albeit with a channel dimension equal to the embedding dimension. We process the BEV tensor by first dividing it into a sequence of flattened BEV patches, i.e., a $N \times P^2D$ tensor, where D denotes the embedding dimension, (P, P) represents the resolution of each image patch, and $N = HW/P^2$ is the total number of BEV patches. The flattened BEV features are then passed through a trainable linear projection to obtain an $N \times D$ patch embedding for each scene. This serves as the equivalent of an N-length input sequence for a Transformer, allowing for attention mechanisms to be applied.

In this first strategy for incorporating BEV features in behavior prediction, visualized in Fig. 2, we alter the modeling of agent-lane interactions. We select BEV grid patches that correspond to agent positions and attend to every other patch in the scene, providing an understanding of the environment surrounding the vehicle. Formally,

$$\mathbf{e}_A = \text{MHA}(Q_A, K_M, V_M), \tag{3}$$

where MHA denotes multi-headed attention, $Q_A \in \mathbb{R}^{M \times D}$ denotes agent patches (queries), $K_M = V_M \in \mathbb{R}^{N \times D}$ denote all map patches (keys and values), and $\mathbf{e}_A$ are the resulting agent-BEV embeddings. As an additional benefit, by computing attention with agent patches, the computational complexity is only $M \times N$ operations, where M denotes the number of agents (most commonly, $M \ll N$).

To verify our approach, we modify the state-of-the-art Transformer-based prediction model HiVT [40]. It employs a hierarchical Transformer structure with a low-level Transformer that encodes agent-lane interactions within a local

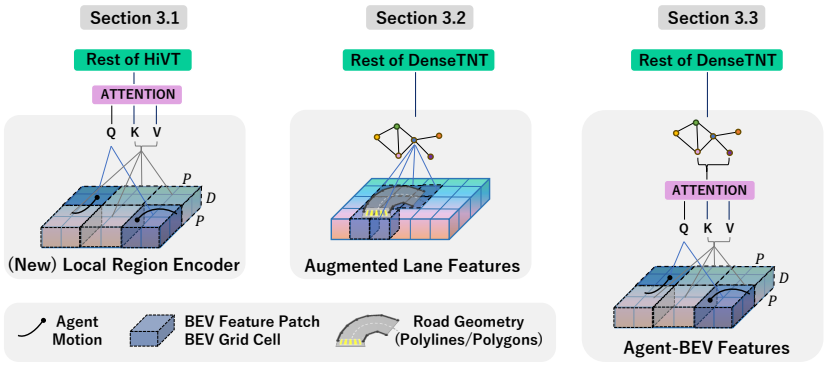

Fig. 2: Three different strategies for incorporating BEV features in behavior prediction. **Left:** local region attention to encode agent-map interaction; **Middle:** augmenting lane vertices with BEV features; **Right:** replacing agent trajectories with temporal BEV features.

neighborhood, followed by a high-level Transformer that models global interactions across the entire scene. Note that, by replacing the local agent-lane interaction encoder with the agent-BEV attention in Eq. (3), we completely remove the use of vectorized lane information in HiVT. The agent-BEV features are then concatenated with agent-agent interactions (modeled via self-attention in HiVT) followed by a linear projection for the global interaction module to process. As we will show in Sec. 4, the prediction performance of HiVT and overall system runtime are greatly enhanced by directly encoding BEV features.

3.2 Augmenting Estimated Lanes with BEV Features

While Sec. 3.1 completely substitutes vectorized map data, another strategy is to augment existing lane information with BEV features (e.g., via concatenation). We first refine the BEV features to match the dimensionality of the latent space associated with raw lane information using a one-dimensional CNN. We then determine the BEV grid positions corresponding to the locations of each map vertex and concatenate the original vertex features (i.e., their positions) with their corresponding BEV features. In doing so, we aim to provide a more comprehensive summary of lane information to downstream modules.

In our work, we instantiate this strategy with a combination of the MapTR line of work [22,23] and DenseTNT [12]. DenseTNT, in particular, is emblematic of prediction models that are heavily map-dependent. It requires lane information at virtually every stage of its pipeline, from initial sparse context encoding to end-point sampling and scoring to guiding predictions during decoding. With such a heavy reliance on vector maps, we cannot eliminate the use of lanes completely as in Sec. 3.1. Instead, we focus on enriching the estimated map vertices by incorporating their corresponding BEV features in DenseTNT's input layer. Specifically, the enriched map elements are encoded using a VectorNet [8] backbone which we augment with a larger layer size to accommodate the increase

in feature dimensionality (full details can be found in the appendix). As we will show in Sec. 4, incorporating BEV features in this manner significantly improves the performance of the associated behavior predictor.

3.3 Replacing Agent Information with Temporal BEV Features

Operating from streaming inputs is a common requirement of online mapping methods deployed on embedded devices. Methods like StreamMapNet [37] introduce a memory buffer to preserve query data and BEV features from previous frames, combining them with the BEV features acquired in the current frame. This introduction of temporal information into the BEV representation enables our third strategy for incorporating BEV features in behavior prediction: replacing agent information with their corresponding BEV features.

In prediction models, agent trajectories are commonly the only source of temporal information during scene context encoding. Vectorized HD maps provide a static understanding of the scene, with fixed road geometry and semantics. While this static-dynamic separation is explicitly handled by prediction architectures, StreamMapNet's approach captures temporal information with a one-step historical BEV feature fusion, enabling it to also capture information about dynamic agents. To leverage StreamMapNet's encoding of both static *and* dynamic information, we additionally modify DenseTNT [12] to replace the agent subgraphs encoded in VectorNet [8] with the agent-BEV features obtained using the attention mechanism in Eq. (3). In doing so, agent trajectory information is completely discarded in DenseTNT [12]. Even so, as we will show in Sec. 4, DenseTNT [12] is able to leverage the implicit trajectory information encoded in the dynamic BEV features and predicts significantly more accurate trajectories.

4 Experiments

Dataset. We evaluate our method on the large-scale nuScenes dataset [1], which includes ground truth (GT) HD maps, multi-sensor data, and tracked agent trajectories. It consists of 1000 driving scenarios with sensor data recorded at 10 Hz and annotated at 2 Hz (i.e., every 5th frame is a keyframe), and is divided into train, validation, and test sets with 500, 200, and 150 scenarios, respectively.

Our work leverages the unified trajdata [17] interface to standardize the data representation between vectorized map estimation models and downstream prediction models. To ensure compatibility across various prediction and mapping models, we upsample the nuScenes trajectory data frequency to 10Hz (matching the sensor frequency) using trajdata's temporal interpolation utilities. Each prediction model is then tasked with forecasting vehicle motion 3 seconds into the future using observations from the preceding 2 seconds.

Metrics. To evaluate trajectory prediction performance, we adopt standard evaluation metrics used in many recent prediction challenges [1, 3, 7, 35]: minimum Average Displacement Error (minADE), minimum Final Displacement Error (minFDE) and Miss Rate (MR). Specifically, minADE evaluates the average Euclidean (ℓ_2) distance between the most-accurately predicted trajectory

Prediction Method	HiVT [40]			DenseTNT [12]		
Online HD Map Method	minADE ↓	minFDE ↓	MR ↓	minADE ↓	minFDE ↓	MR ↓
MapTR [22]	0.4234	0.8900	0.0955	1.0462	2.0661	0.3494
MapTR [22] + Unc [13]	0.4036	0.8372	0.0822	1.1190	2.1502	0.3669
MapTR [22] + Ours	**0.3617** (−15%)	**0.7401** (−17%)	**0.0720** (−25%)	**0.7608** (−27%)	**1.4700** (−29%)	**0.2593** (−26%)
MapTRv2 [23]	0.3950	0.8310	0.0894	1.2648	2.3481	0.4043
MapTRv2 [23] + Unc [13]	0.3896	0.8085	0.0859	1.3228	2.4821	0.4406
MapTRv2 [23] + Ours	**0.3844** (−3%)	**0.7848** (−6%)	**0.0741** (−17%)	**1.1232** (−11%)	**2.3000** (−2%)	**0.4025** (0%)
MapTRv2-CL [23]	0.3657	0.7473	0.0710	0.7664	**1.3174**	**0.1547**
MapTRv2-CL [23] + Unc [13]	**0.3588**	**0.7232**	**0.0660**	0.8123	1.3426	0.1567
MapTRv2-CL [23] + Ours	0.3652 (0%)	0.7323 (−2%)	0.0710 (0%)	**0.7630** (0%)	1.3609 (+3%)	0.1576 (+2%)
StreamMapNet [37]	0.4035	0.8569	0.0996	0.8864	1.7050	0.2467
StreamMapNet [37] + Unc [13]	0.3907	0.8034	0.0812	0.9220	1.6851	0.2310
StreamMapNet [37] + Ours	**0.3800** (−6%)	**0.7709** (−10%)	**0.0746** (−25%)	**0.7377** (−17%)	**1.3661** (−20%)	**0.1987** (−19%)

Table 1: Virtually every combination of mapping and prediction benefits from directly leveraging upstream BEV features on the nuScenes [1] dataset, with certain combinations achieving performance improvements of 25% or more. Percent values denote the relative improvement in prediction performance achieved by our approach.

(of 6 predicted trajectories) and the GT trajectory across the prediction horizon. minFDE measures the ℓ_2 distance between the end point of these two trajectories. MR refers to the proportion of predicted trajectories that have an FDE of more than 2 meters from the GT endpoint.

Models and Training. To evaluate the effect of incorporating BEV features in downstream prediction models, we train DenseTNT [12] and HiVT [40] on the outputs of four online mapping models (MapTR [22], MapTRv2 [23], MapTRv2 with Centerlines [23], and StreamMapNet [37]) in one of three setups: Baseline (using vectorized inputs), Uncertainty-enhanced (where each map element vertex contains spatial uncertainty information) [13], and Ours (one of the BEV feature attention strategies detailed in Sec. 3), yielding a total of 24 model combinations.

In particular, we first train the four online map estimation models to convergence following the models' original training recipes in the baseline setup or the training recipes of [13] in the uncertainty-enhanced setup. We then extract BEV features from each model and scene, and train behavior prediction models according to the above three setups: using only vectorized map information as a baseline, leveraging uncertainty as in [13], and our approach of leveraging BEV features as in Sec. 3. All models are trained on a single RTX4090 GPU. Full model and training details can be found in the appendix.

4.1 Leveraging BEV Features in Behavior Prediction

Prediction Accuracy Improvements. As shown in Tab. 1, for virtually all mapping/prediction model combinations, incorporating BEV features leads to *significantly* better prediction accuracy, not only compared to the baseline models but also to the uncertainty-enhanced approach. The largest improvements (up to 25% and more) are in MR and minFDE, suggesting that latent BEV features can especially help with long-horizon prediction performance. Endpoint prediction accuracy is especially important for trajectory prediction as it directly impacts later planning accuracy.

Note that, while MapTR [22] does not perform as well as its successor (MapTRv2 [23]) or the temporally-enhanced StreamMapNet [37], its BEV features

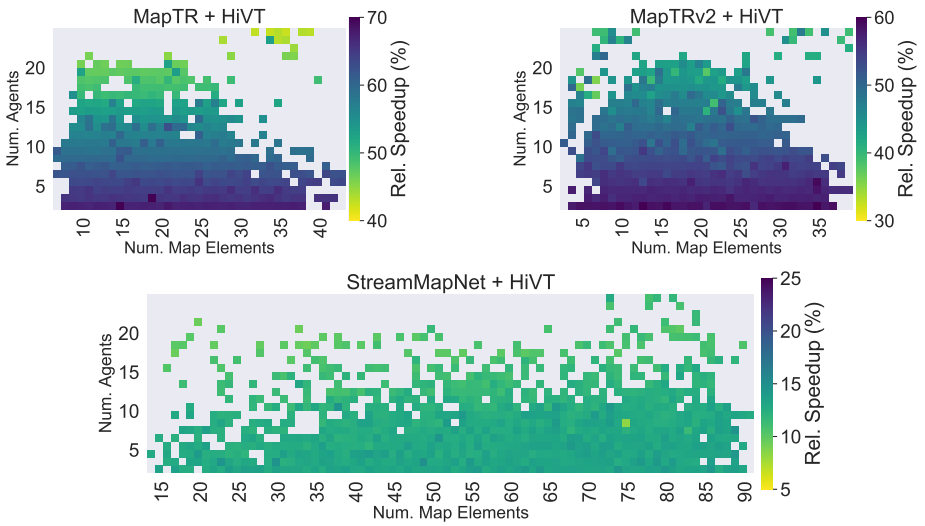

Fig. 3: Our integrated BEV-prediction approach runs faster than decoupled baselines across all scenario sizes (number of agents and map elements) and mapping models.

provide the largest improvement to downstream performance, yielding prediction accuracy that outperforms combinations with MapTRv2 [23] (without centerlines). This result suggests that MapTR's decoder may be introducing unwanted noise, which impedes its ability to generate precise map elements. Accordingly, by leveraging information from earlier stages of MapTR [22], both HiVT [40] and DenseTNT [12] obtain significant improvements in prediction performance, emphasizing the benefits of deeper integrations between mapping and prediction through an intermediate BEV representation.

The additional production of centerlines in MapTRv2-CL [23] yields the most accurate predictions overall. Accordingly, the benefits of incorporating BEV features are the least pronounced. This result reaffirms the utility of centerlines for trajectory prediction and provides guidance for future map estimation research regarding which type of map element is most useful.

Finally, although StreamMapNet's BEV features are leveraged in two completely different ways (as a substitute for lane information in HiVT [40] and as a substitute for agent information in DenseTNT [12]), they both provide significant improvements to the baseline prediction models. This indicates that temporal information not only helps in decoding more accurate maps, it also provides a temporal understanding of the behavior of agents which is particularly valuable for trajectory prediction.

Inference Speedup. In Fig. 3, we compare the GPU inference speedup achieved by our integrated approach (described in Sec. 3.1) relative to decoupled baselines for HiVT [40]. For both approaches, runtime is measured starting from the processing of the input RGB images and ending at the output of the final trajectories. As can be seen, our approach results in significant inference time improvements due to its elimination of the time-consuming map decoding stage. Specifically, our integrated method is 42-73% faster than MapTR [22] and HiVT

[40] alone, 35-62% faster than MapTRv2 [23] and HiVT [40] alone, and 8-15% faster than StreamMapNet [37] and HiVT [40] alone. Identical improvements are obtained when compared to the uncertainty-integrated method [13], reaffirming the strength of our approach.

Further, the overall inference time of the baseline mapping and prediction models scales with both the number of map elements and number of agents present in a scene. In contrast, our integrated approach is much less sensitive to the number of map elements, and this is reflected in Fig. 3 where our approach's runtime improves more as the number of map elements increases. This occurs because each agent's patch attends to every other BEV *patch* in Sec. 3.1. Thus, even if the number of map elements is high, the number of BEV patches is fixed, significantly reducing inference time by eliminating the need to process map elements through HiVT's encoder.

In scenarios where the number of map elements is low but the number of agents is high (top left of Fig. 3), the reduction in processing time is less pronounced. Given the smaller quantity of map elements, HiVT [40] naturally requires less time to encode them. Further, with an increase in the number of agents, the approach in Sec. 3.1 must perform additional attention operations as each new agent introduces an extra N attention operations. This increase in computation partially offsets the savings achieved by not processing vectorized map elements, leading to smaller improvements in run time. Nevertheless, our approach still yields substantial reductions in inference time.

4.2 Ablation Studies

Patch Size. Tab. 2 illustrates the effect of BEV feature patch size P on prediction performance. A patch size that is too small results in insufficient information capture. For instance, MapTRv2 operates with a $60m \times 30m$ perception range and has a BEV dimension ($H \times W$) of 200×100, meaning each BEV grid cell represents a $0.3m \times 0.3m$ square in the real world. This size is relatively small, particularly as we wish to capture global information in the scene. As shown in Tab. 2, a patch size of 10×5 (covering $3m \times 1.5m$) underperforms compared to a more moderately-sized 20×20 patch ($6m \times 6m$). Larger patch sizes do not yield performance improvements, however, likely due to the loss of granular information when converting patches into smaller vector embeddings (projecting from P^2D to D dimensions), which also inhibits prediction accuracy as reflected by worsening minFDE and MR values. Overall, we find that a patch size of 20×20 yields the best prediction performance.

BEV Encoder Selection. As mentioned in Sec. 2, map estimation models typically employ one of two distinct PV2BEV encoders: BEVFormer [20], which is used in MapTR [22] and StreamMapNet [37], and LSS [28], used in MapTRv2 [23] and MapTRv2-Centerline [23]. As described in Sec. 3, BEVFormer [20] enhances its BEV features by integrating historical BEV features (B_{t-1}) through temporal self-attention. In contrast, LSS [28] processes only data from the current frame. This divergence in backbone mechanisms leads to notable performance disparities between MapTR [22] and its subsequent deriva-

Models	MapTRv2-Centerline [23] and HiVT [40]		
Patch Sizes	minADE ↓	minFDE ↓	MR ↓
(10, 5)	0.3845	0.8003	0.0853
(20, 10)	0.3729	0.7616	0.0749
(20, 20)	0.3728	**0.7518**	**0.0737**
(20, 25)	**0.3709**	0.7649	0.0761
(40, 20)	0.3737	0.7583	0.0770

Table 2: An exploration of BEV patch sizes reveals that there are detriments to having patches that are too small (insufficient information capture) or too large (granular information loss), with the best performance achieved by a patch size of 20×20 (corresponding to $6m \times 6m$ in the real world).

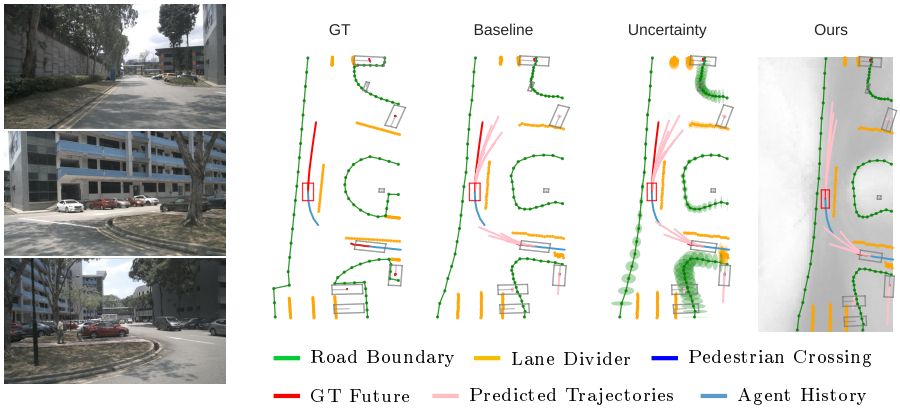

Fig. 4: StreamMapNet [37] and HiVT [40] combined using the strategy in Sec. 3.1. By replacing lane information with temporal BEV features, HiVT is able to keep its predicted trajectories in the current lane, closely aligning with the GT trajectory.

tives MapTRv2 [23] and MapTRv2-Centerline [23]. Despite the use of a similar decoding mechanism (a hierarchical query embedding scheme followed by hierarchical bipartite matching), the utilization of an LSS-based BEV extractor results in a lack of temporal information in MapTRv2's resulting BEV features. This results in relative inefficiency within our methodology, showcasing only modest enhancements when compared against integrations with MapTR [22] and StreamMapNet [37].

This divergence is empirically observed in Tab. 1, where MapTRv2 and MapTRv2-Centerline's BEV features (which only encode static information) yield a mere relative improvement of 4%, 2% and 4% in minADE, minFDE, and MR, respectively. In comparison, the use of temporally-informed BEV features in MapTR [22] and StreamMapNet [37] yields relative performance enhancements of 16%, 19%, and 24% in minADE, minFDE, and MR for both mapping/prediction model combinations. These improvements underscore the utility of integrating temporal dynamics into BEV features for improved trajectory forecasting.

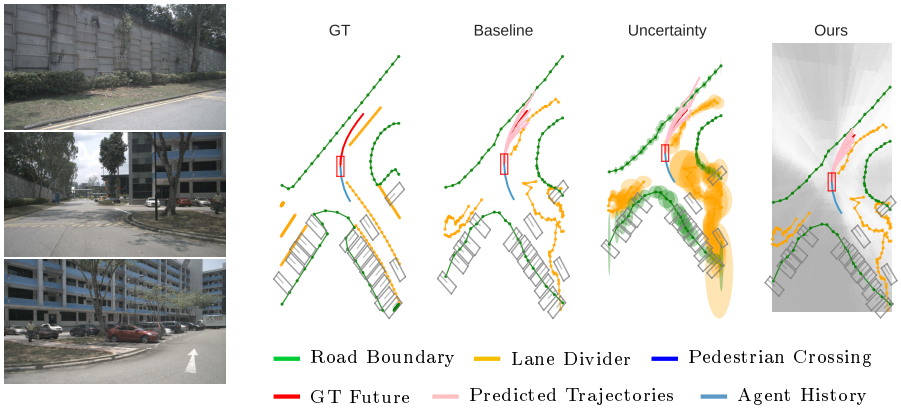

Fig. 5: MapTR [22] and DenseTNT [12] combined via the strategy in Sec. 3.2. Our augmentation of map vertices with BEV features enables DenseTNT to produce very accurate trajectories, preventing the road boundary incursions seen in the Baseline and Uncertainty-enhanced [13] setups.

4.3 Qualitative Comparisons

BEV Feature Visualization. Aside from estimated maps and their uncertainties, we also visualize the corresponding BEV features for each test scene in Figs. 4 to 6. These visualizations are obtained by first reducing the dimensionality of each BEV grid cell to a single value using principal component analysis (PCA), followed by normalization to $[0, 255]$, creating a grayscale image. The resulting BEV feature images form the background of our method plots.

In instances where SteamMapNet [37] serves as the mapping model (Figs. 4 and 6), a distinct separation is observed between driveable areas (gray) and the non-driveable areas (white) beyond the designated boundaries. This distinction highlights the comprehensive geometric information captured within the temporal BEV features of SteamMapNet [37], enabling the implicit modeling of map characteristics from BEV features alone. This interpretation is inverted in MapTR's BEV features (Fig. 5), white indicates driveable areas and gray signifies non-driveable regions. In both cases, BEV features play a critical role in informing future predictions.

Fig. 4 visualizes a parking lot next to a building. Utilizing the temporal BEV features from StreamMapNet [37], HiVT [40] is able to have all six predicted trajectories tightly clustered around the GT. This precision allows HiVT [40] to maintain lane discipline, avoiding encroaching into opposing lanes (seen in the Baseline and Uncertainty setups).

In Fig. 5, MapTR's estimated map vertices are augmented with their corresponding BEV features. Specifically, the grey area depicted ahead of the center vehicle covers part of the road boundary, which provides an additional non-driveable signal and reinforces the presence of a road border. By leveraging this extra information, DenseTNT [12] much more effectively restricts the vehicle's trajectories from crossing into the non-drivable area. In contrast, both the Baseline and Uncertainty-enhanced approaches produce trajectories that inter-

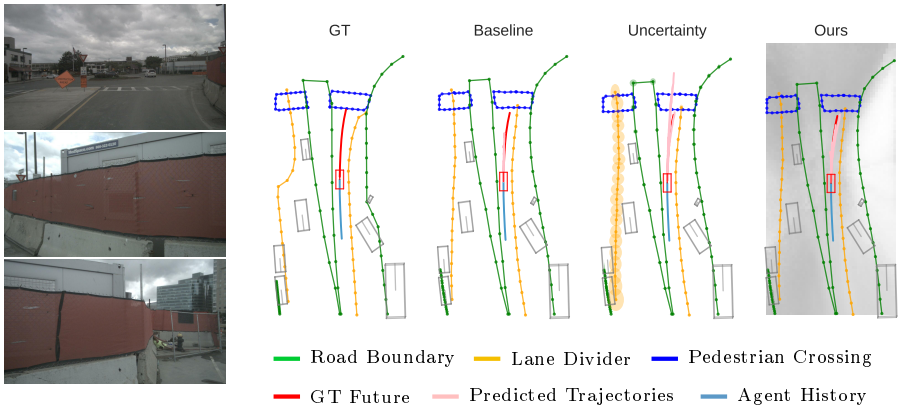

Fig. 6: StreamMapNet [37] and DenseTNT [12] combined using the strategy in Sec. 3.3. By replacing agent trajectory information with BEV features, DenseTNT is able to predict trajectories that stop before the crosswalk, compared to the undershooting and overshooting of the Baseline and Uncertainty-enhanced [13] approaches.

sect with the road boundary and lane divider. By incorporating BEV features, DenseTNT [12] is able to confine its future predictions to the designated white, driveable area, showing the utility of BEV features in improving map adherence.

In Fig. 6, the Baseline predicted trajectories undershoot the GT, whereas the Uncertainty-enhanced approach overshoots the pedestrian crosswalk entirely. By encoding the BEV features directly for agent information, our approach strikes a balance and produces trajectories that align well with the GT trajectory.

5 Conclusion

In this work, we propose three different strategies to leverage the intermediate BEV features within online map estimation models in downstream tasks such as behavior prediction. We systematically evaluate the benefits of different BEV encoding strategies and demonstrate how incorporating BEV features in downstream behavior prediction results in significant performance and runtime improvements. In particular, combinations of various online mapping and prediction methods achieve up to 73% faster inference times when operating directly from intermediate BEV features and produce predictions that are up to 29% more accurate across a variety of evaluation metrics.

Our work's limitations and potential negative impacts relate to its use of black-box features in lieu of vectorized map estimation. While this yields performance and runtime improvements, it may complicate introspection into why a behavior prediction algorithm made certain predictions (compared to when explicit map elements are encoded). Towards this end, exciting future directions include further explorations of mapping models' BEV feature spaces, strategies to interpret BEV features at runtime (alternatives to costly decoding processes), and co-training strategies to inform upstream map estimation models of the task of behavior prediction (ideally yielding improvements in both mapping and prediction performance, towards the development of end-to-end AV stacks).

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: IEEE Conf. on Computer Vision and Pattern Recognition (2020)
2. Can, Y.B., Liniger, A., Paudel, D.P., Gool, L.V.: Structured bird's-eye-view traffic scene understanding from onboard images. In: IEEE Int. Conf. on Computer Vision (2021)
3. Chang, M.F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D., Hays, J.: Argoverse: 3d tracking and forecasting with rich maps. In: IEEE Conf. on Computer Vision and Pattern Recognition (2019)
4. Deo, N., Wolff, E.M., Beijbom, O.: Multimodal trajectory prediction conditioned on lane-graph traversals. In: Conf. on Robot Learning (2021)
5. Dong, H., Zhang, X., Xu, J., Ai, R., Gu, W., Lu, H., Kannala, J., Chen, X.: SuperFusion: Multilevel LiDAR-camera fusion for long-range HD map generation. arXiv preprint arXiv:2211.15656 (2022)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Int. Conf. on Learning Representations (2021)
7. Ettinger, S., Cheng, S., Caine, B., Liu, C., Zhao, H., Pradhan, S., Chai, Y., Sapp, B., Qi, C., Zhou, Y., Yang, Z., Chouard, A., Sun, P., Ngiam, J., Vasudevan, V., McCauley, A., Shlens, J., Anguelov, D.: Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In: IEEE Int. Conf. on Computer Vision (2021)
8. Gao, J., Sun, C., Zhao, H., Shen, Y., Anguelov, D., Li, C., Schmid, C.: VectorNet: Encoding HD maps and agent dynamics from vectorized representation. In: IEEE Conf. on Computer Vision and Pattern Recognition (2020)
9. Gilles, T., Sabatini, S., Tsishkou, D., Stanciulescu, B., Moutarde, F.: HOME: Heatmap output for future motion estimation. In: Proc. IEEE Int. Conf. on Intelligent Transportation Systems (2021)
10. Gilles, T., Sabatini, S., Tsishkou, D., Stanciulescu, B., Moutarde, F.: GOHOME: Graph-oriented heatmap output for future motion estimation. In: Proc. IEEE Conf. on Robotics and Automation (2022)
11. Gilles, T., Sabatini, S., Tsishkou, D., Stanciulescu, B., Moutarde, F.: THOMAS: Trajectory heatmap output with learned multi-agent sampling. In: Int. Conf. on Learning Representations (2022)
12. Gu, J., Sun, C., Zhao, H.: DenseTNT: End-to-end trajectory prediction from dense goal sets. In: IEEE Int. Conf. on Computer Vision (2021)
13. Gu, X., Song, G., Gilitschenski, I., Pavone, M., Ivanovic, B.: Producing and leveraging online map uncertainty in trajectory prediction. In: IEEE Conf. on Computer Vision and Pattern Recognition (2024)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conf. on Computer Vision and Pattern Recognition (2016)
15. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., Lu, L., Jia, X., Liu, Q., Dai, J., Qiao, Y., Li, H.: Planning-oriented autonomous driving. In: IEEE Conf. on Computer Vision and Pattern Recognition (2023)

16. Ivanovic, B., Harrison, J., Pavone, M.: Expanding the deployment envelope of behavior prediction via adaptive meta-learning. In: Proc. IEEE Conf. on Robotics and Automation (2023), https://arxiv.org/abs/2209.11820
17. Ivanovic, B., Song, G., Gilitschenski, I., Pavone, M.: trajdata: A unified interface to multiple human trajectory datasets. In: Conf. on Neural Information Processing Systems Datasets and Benchmarks Track. New Orleans, USA (Dec 2023), https://arxiv.org/abs/2307.13924
18. Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: VAD: Vectorized scene representation for efficient autonomous driving. In: IEEE Int. Conf. on Computer Vision (2023)
19. Li, Q., Wang, Y., Wang, Y., Zhao, H.: HDMapNet: An online HD map construction and evaluation framework. In: Proc. IEEE Conf. on Robotics and Automation (2022)
20. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: European Conf. on Computer Vision (2022)
21. Liang, M., Yang, B., Hu, R., Chen, Y., Liao, R., Feng, S., Urtasun, R.: Learning lane graph representations for motion forecasting. In: European Conf. on Computer Vision (2020)
22. Liao, B., Chen, S., Wang, X., Cheng, T., Zhang, Q., Liu, W., Huang, C.: MapTR: Structured modeling and learning for online vectorized HD map construction. In: Int. Conf. on Learning Representations (2023)
23. Liao, B., Chen, S., Zhang, Y., Jiang, B., Zhang, Q., Liu, W., Huang, C., Wang, X.: MapTRv2: An end-to-end framework for online vectorized HD map construction. arXiv preprint arXiv:2308.05736 (2023)
24. Liu, Y., Yuantian, Y., Wang, Y., Wang, Y., Zhao, H.: VectorMapNet: End-to-end vectorized HD map learning. In: Int. Conf. on Machine Learning. PMLR (2023)
25. Liu, Y., Zhang, J., Fang, L., Jiang, Q., Zhou, B.: Multimodal motion prediction with stacked transformers. In: IEEE Conf. on Computer Vision and Pattern Recognition (2021)
26. Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D., Han, S.: BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In: Proc. IEEE Conf. on Robotics and Automation (2023)
27. Phan-Minh, T., Grigore, E.C., Boulton, F.A., Beijbom, O., Wolff, E.M.: CoverNet: Multimodal behavior prediction using trajectory sets. In: IEEE Conf. on Computer Vision and Pattern Recognition (2020)
28. Philion, J., Fidler, S.: Lift, Splat, Shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. In: European Conf. on Computer Vision (2020)
29. Qiao, L., Ding, W., Qiu, X., Zhang, C.: End-to-end vectorized HD-map construction with piecewise bezier curve. In: IEEE Conf. on Computer Vision and Pattern Recognition (2023)
30. Rudenko, A., Palmieri, L., Herman, M., Kitani, K.M., Gavrila, D.M., Arras, K.O.: Human motion trajectory prediction: A survey. Int. Journal of Robotics Research **39**(8), 895–935 (2020)
31. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In: European Conf. on Computer Vision (2020), https://arxiv.org/abs/2001.03093
32. Shin, J., Rameau, F., Jeong, H., Kum, D.: InstaGraM: Instance-level graph modeling for vectorized HD map learning. arXiv preprint arXiv:2301.04470 (2023)

33. Tong, W., Sima, C., Wang, T., Chen, L., Wu, S., Deng, H., Gu, Y., Lu, L., Luo, P., Lin, D., Li, H.: Scene as occupancy. In: IEEE Int. Conf. on Computer Vision (2023)
34. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Conf. on Neural Information Processing Systems (2017)
35. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., Ramanan, D., Carr, P., Hays, J.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. In: Conf. on Neural Information Processing Systems Datasets and Benchmarks Track (2021)
36. Xu, Z., Wong, K.K., Zhao, H.: InsightMapper: A closer look at inner-instance information for vectorized high-definition mapping. arXiv preprint arXiv:2308.08543 (2023)
37. Yuan, T., Liu, Y., Wang, Y., Wang, Y., Zhao, H.: StreamMapNet: Streaming mapping network for vectorized online HD map construction. In: IEEE Winter Conf. on Applications of Computer Vision (2024)
38. Yuan, Y., Weng, X., Ou, Y., Kitani, K.M.: AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting. In: IEEE Int. Conf. on Computer Vision. pp. 9813–9823 (2021)
39. Zhao, H., Gao, J., Lan, T., Sun, C., Sapp, B., Varadarajan, B., Shen, Y., Shen, Y., Chai, Y., Schmid, C., Li, C., Anguelov, D.: TNT: Target-driveN Trajectory Prediction. In: Conf. on Robot Learning (2020)
40. Zhou, Z., Ye, L., Wang, J., Wu, K., Lu, K.: HiVT: Hierarchical vector transformer for multi-agent motion prediction. In: IEEE Conf. on Computer Vision and Pattern Recognition (2022)