MapPrior: Bird's-Eye View Map Layout Estimation with Generative Models

Xiyue Zhu[1] Vlas Zyrianov[1] Zhijian Liu[2] Shenlong Wang[1]
[1]University of Illinois at Urbana-Champaign [2]MIT

https://mapprior.github.io

Abstract

Despite tremendous advancements in bird's-eye view (BEV) perception, existing models fall short in generating realistic and coherent semantic map layouts, and they fail to account for uncertainties arising from partial sensor information (such as occlusion or limited coverage). In this work, we introduce ***MapPrior****, a novel BEV perception framework that combines a traditional discriminative BEV perception model with a learned generative model for semantic map layouts. Our MapPrior delivers predictions with better* ***accuracy****,* ***realism*** *and* ***uncertainty awareness****. We evaluate our model on the large-scale nuScenes benchmark. At the time of submission, MapPrior outperforms the strongest competing method, with significantly improved MMD and ECE scores in camera- and LiDAR-based BEV perception. Furthermore, our method can be used to perpetually generate layouts with unconditional sampling.*

1. Introduction

Accurately understanding the surrounding environment of autonomous vehicles is crucial to guarantee the safety of riders and other traffic participants. Among various perception approaches, Bird's-Eye View (BEV) perception has drawn significant attention in recent years thanks to its capacity to densely model scene layouts and its tight coupling with downstream planning [80, 56, 49].

Existing perception models encounter two challenges. The first challenge pertains to the limitations of observations, particularly in distant or occluded regions, resulting in inaccurate predictions. This may manifest in choppy, out-of-distribution, or missing map elements. The second challenge is that most existing models do not consider uncertainty and diversity in possible road layouts. Taking Fig. 1 as an example, the state-of-the-art LiDAR perception model [80] generates incoherent lane markings and sidewalks with massive gaps and cannot quantify the uncertainty and multi-modality as it only produces a single prediction per input.

This paper presents MapPrior, a novel BEV perception method that is accurate, realistic, and uncertainty-aware. At the heart of our method is a novel combination of the standard discriminative BEV perception model with a learned deep generative traffic layout prior. Incorporating generative modeling in this predictive task attempts to address the two aforementioned challenges – modeling the data distribution improves realism, and using a sampling process allows generating multiple realistic predictions. Combining our generative model with a discriminative perception model ensures that our method retains a strong predictive ability.

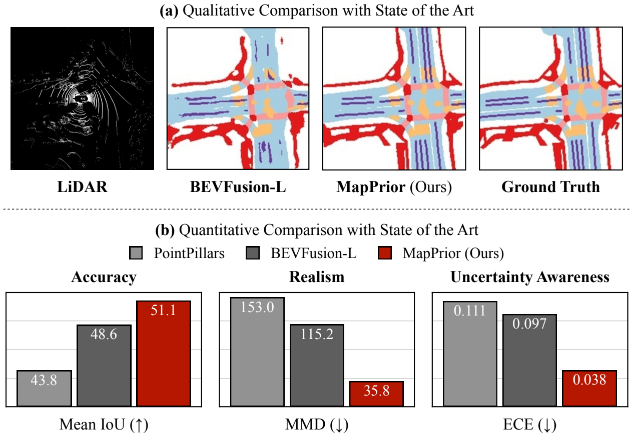

Figure 1: Existing predictive BEV perception models do not provide realistic scene structures (*e.g*., the lane and sidewalks have gaps and are not straight). In contrast, our MapPrior is able to accurately recover the layout with a learned prior. These results are validated with a quantitative comparison, showing our method's superior accuracy, realism and uncertainty awareness.

Our approach comprises two steps, namely the prediction and generative steps. In the prediction step, we use an off-the-shelf BEV perception model [49] to make an initial layout estimate of the traffic scene from sensory input. In the generative step, we use our MapPrior model and initial estimate to sample one or multiple refined layouts. We perform sampling in a learned discrete latent space using a conditional transformer that is provided with the initial prediction. Finally, the generated tokens are passed into a decoder to output the final layout prediction, which is diverse, realistic, and coherent with the input. The encoder, decoder, and codebook of the MapPrior are trained from real-world map data in an unsupervised way.

We benchmark our method on the nuScenes dataset against various state-of-the-art BEV perception methods with varying modalities. Our results show that MapPrior outperforms existing methods in terms of accuracy (as reflected by mean intersection-over-union), realism (as reflected by maximum-mean discrepancy), and uncertainty awareness (as reflected by expected calibration error). Furthermore, we demonstrate the unconditional generation capabilities of MapPrior by generating a realistic and consistent HD map of a 30 km-long road.

2. Related Work

Self-driving perception aims to interpret sensory input and establish a representation of the surrounding environment from onboard sensors. It is a critical component for ensuring the safety and efficiency of autonomous vehicles, and various methods have been proposed to improve accuracy and robustness [71, 39]. Traditional approaches for self-driving perception involve multiple (isolated) subtasks, such as localizing against a pre-scanned environment [40, 4], object detection and tracking [38, 80, 68, 44, 50], image segmentation [62, 85, 10] and trajectory prediction [23, 64, 84, 43, 7, 82].

Recently, there has been a growing interest in a unified Bird's Eye View (BEV) perception, both in academia [60, 87, 56, 76, 8, 50] and industry [26]. This approach aims to produce a top-down semantic layout of the traffic scene from sensory input, which is efficient, informative, and compact. Notably, the top-down layout is closely linked to downstream 2D motion planning for wheeled robots, making BEV perception particularly suitable for self-driving navigation.

Various BEV perception modules have been studied in robotics and computer science to improve perception accuracy. These approaches typically take LiDAR [80, 38, 51, 67], multi-view camera [60, 87, 56, 76, 11, 46, 26, 59, 21], or both [81, 75, 49, 2, 57] as input, and output segmentations for road elements and detections for dynamic traffic participants like cars, cyclists or pedestrians. The majority of these methods are supervised and rely on predictive networks like CNNs [59, 67, 80, 38, 44, 60, 56] and transformers [2, 57, 11, 46, 51, 87, 21, 26, 27]. Despite their success, there are still unresolved challenges. For example, real-world traffic layouts are highly structured, with straight lane marks, strong topological relationships, and sharp road element boundaries. But such structures have proven difficult to preserve in the BEV perception output map, even with state-of-the-art methods, as shown in Fig. 1. This can harm the realism of perception output and the practicality of the resulting map for motion planning. Additionally, most networks make a single layout estimation without diversity or calibrated uncertainty, leaving the autonomy vulnerable to catastrophic failure.

HDMapNet [42], VectorMapNet [47], and others [52, 8] have been proposed to address the issue of layout structure by introducing a vectorized output map format with implicitly structured priors. However, this approach fails to account for all layer types and still produces significant artifacts in regions with limited observations. Very recently, calibrated confidence scores have been investigated to address the aforementioned uncertainty issue [37, 13], but the inherent multi-modal uncertainty in BEV perception remains under-explored.

In contrast, our approach leverages generative modeling as a map prior, which encodes the rich structure of the traffic scene. Our conditional generative modeling also allows us to sample multiple diverse outputs for the same input. Consequently, our results are more accurate and realistic with better multi-modal uncertainty modeling.

Generative models. Our approach to generative BEV perception involves learning a map prior and performing conditional sampling using deep generative models. Generative models learn to capture the underlying structure of the data distribution and create new, diverse samples. Several well-known approaches include VAEs (variational autoencoders) [34, 73], which use variational inference to learn a latent space model, and autoregressive models [55], which decompose the generation problem into simpler conditional generation tasks. GANs (generative adversarial networks) [20, 83, 1, 31, 28, 72, 5] use an adversarial loss to train networks to convert noise into samples. Flow-based models [33, 15] exploit an invertible process to sample from a proposal distribution. Recently, diffusion models [14, 24] have been developed that generate samples through a denoising-diffusion process. Our method leverages vector-quantized generative models [74, 18, 9], which use a discrete-valued latent space representation that is powerful, structure-preserving, and efficient for conditional sampling. Specifically, we build our map prior on top of this approach to capture the discrete structure of the map data and generate high-quality samples.

Generative approach for vision. Our approach belongs to the broader category of generative modeling approaches in computer vision [79, 53, 58, 69], which includes Markov random fields (MRFs), factor graphs, energy-based models, deep generative models, among others. These practices date back to the 1970s. In contrast to discriminative or predictive approaches, generative approaches often tackle the task

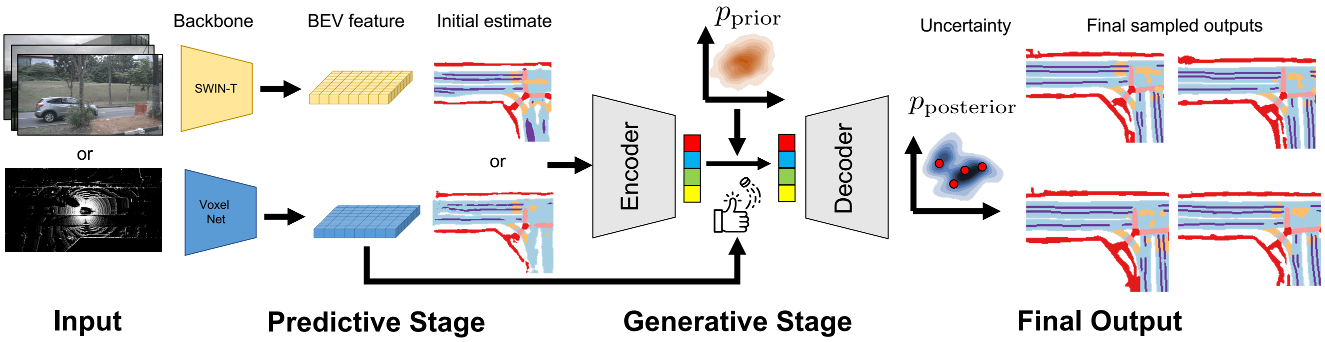

Figure 2: MapPrior first makes use of an off-the-shelf perception model to generate an initial noisy estimate from the sensory input. It then encodes the noisy estimate into a discrete latent code using a generative encoder and generates various samples through a transformer-based controlled synthesis. Finally, MapPrior decodes these samples into outputs with a decoder.

as a marginal sampling or MAP inference problem, which is more effective at utilizing strong prior knowledge and modeling uncertainty. Representative works can be seen in low-level vision [19, 89], optical flow [70, 63], and image segmentation [66, 35, 86]. Generative approaches with conditional sampling have often been investigated in vision tasks involving multimodality or uncertainty, such as image editing [88, 28, 12, 54], image segmentation [41] and stereo estimation [16, 77]. However, few studies have explored their use in self-driving perception, despite its inherent multimodal uncertainty nature.

3. Approach

The objective of this research is to develop a method that can generate a *precise*, *realistic*, and *uncertainty-aware* map layout from sensory input in one or a few modalities. To achieve this, we introduce a new framework named MapPrior that combines the predictive capability of discriminative models with the capacity of generative models to capture structures and uncertainties. We present a two-stage conditional sampling framework to explain our approach and detail the implementation of each module. Additionally, we discuss our design decisions, the learning process, and how our approach relates to previous methods.

3.1. Overview

Formulation. We formulate the probabilistic BEV perception problem as a conditional sampling task. Given the sensory input $\mathbf{x} \in \mathcal{X}$ (which could be from a camera, LiDAR or multiple sensors), we aim to find one or multiple plausible traffic layouts $\mathbf{y} \in \mathcal{Y}$ from the top-down bird's-eye view. The traffic layout $\mathbf{y}$ is a multi-layer binary map centered at the ego vehicle. Conventional methods [49, 80, 56, 81, 76] rely on a deterministic predictive network $\mathbf{y} = f_\theta(\mathbf{x})$ to provide a single output estimation. We propose, instead, to use a generative vector-quantized latent space model to model the uncertainty and diversity. Specifically, we aim to design a conditional probability model $p_\theta(\mathbf{y}|\mathbf{x})$ that can sample our desired output $\mathbf{y}$ from a distribution. This allows us to explicitly model uncertainty and generate multiple plausible traffic layouts.

Motivation. Intuitively speaking, the objective of a predictive perception model is to produce a coherent semantic layout that best represents the sensory input. However, this training objective does not necessarily optimize for structure preservation or realism. For example, minor defects in lane markers might not significantly affect cross-entropy, but could drastically impact downstream modules due to topological changes. On the other hand, a generative prior model, such as a GAN, is trained to capture realism in structures. The advantage of such a model is that it can be trained on HD map collections, without paired data, in an unsupervised manner. This insight inspires our proposed solution, which combines a predictive model with a generative model to tackle the conditional sampling task with both coherence and realism in mind.

3.2. Inference

In our framework, the distribution $p_\theta(\mathbf{y}|\mathbf{x})$ is defined implicitly using a latent model $\mathbf{z}$. The framework comprises of a predictive stage and a generative stage. During the predictive stage, the perception module $F(\mathbf{x})$ generates an initial noisy estimate $\mathbf{y}'$ for input $\mathbf{x}$. For the generative stage, we take inspiration from recent successful techniques in vector-quantized generation [18, 9, 61] for conditional sampling and use a VQGAN-like model to generate multiple realistic samples. To achieve this, we encode the noisy estimate $\mathbf{y}'$ into a discrete latent code $\mathbf{z}'$ using a generative encoder $E(\mathbf{y}')$. This latent code $\mathbf{z}'$ and the sensory input $\mathbf{x}$ then guide the generative synthesis process through a transformer-based controlled synthesis in the latent space, producing various samples $\{\mathbf{z}^{(k)} \sim p(\mathbf{z}|\mathbf{z}', \mathbf{x})\}$. Finally, these samples are decoded into multiple output samples using a decoder $G(\cdot)$: $\mathbf{y}^{(k)} = G(\mathbf{z}^{(k)})$, which provides our final layout estimation samples. Fig. 2 depicts the overall inference framework.

Predictive stage. The predictive stage aims to establish reliable initial layout estimation that can act as a guiding control during the conditional sampling stage. To achieve this, we have incorporated a predictive sensory backbone, VoxelNet [78, 80] for LiDAR and Swin Transformer [48] for multi-view camera inputs. The sensory backbone networks first extract features from the sensory space and then project them into bird's eye view features. For the LiDAR input, a 3D convolution backbone is utilized as a feature extractor, and the LiDAR features are flattened along the height dimension to project them to BEV. In contrast, for the camera input, a hierarchical transformer is used as a feature extractor for each view, and monocular depth estimation and view transformation are applied to project the features to BEV. The predictive model employs a convolutional segmentation head to generate the layout estimates, denoted by $\mathbf{y}' = F(\mathbf{x})$. It should be noted that the resulting BEV map achieves a reasonable intersection-over-union score (IoU). However, as Fig. 1 highlights, the model suffers from unrealistic structures, missing road elements, and noise, particularly in regions with limited observations.

Generative stage. To enhance the quality and diversity of our perception, we incorporate a generative map prior in the second stage for conditional generation. The generative map prior is built on a VQGAN architecture [18], which consists of three learnable components: the encoder E, the decoder G, and the codebook $\mathcal{C} = \{\mathbf{c}_j\}$ with j being the code index. The encoder transforms a traffic layout into a latent feature map, where each element of the spatial map is chosen from one of the codes in the codebook as follows:

$$\mathbf{z}_t(\mathbf{y}) = \underset{\mathbf{c}_j \in \mathcal{Z}}{\arg\max} \, \|\mathbf{c}_j - E_t(\mathbf{y})\|_2^2, \quad \forall t \tag{1}$$

where t is the t-th entry of the feature map $\mathbf{z}$. The decoder then transforms the latent code back to the layout map space: $G(\mathbf{z}(\mathbf{y}))$. Sampling from this prior can be done by randomly drawing a latent code and decoding it into a layout map. The discrete-valued auto-encoder architecture greatly regularizes the output space in a structured manner, preventing it from producing unrealistic reconstructions, as shown in Fig 1.

During conditional sampling at inference, we first encode the noisy estimate $\mathbf{y}'$ into a discrete latent code $\mathbf{z}'$ using a generative encoder E. This latent code $\mathbf{z}'$ and the sensory input $\mathbf{x}$ are then used as guidance by a transformer $T(\mathbf{z}', \mathbf{x})$ for the generative synthesis of the latent space, producing various samples $\mathbf{z}^{(k)} \sim p(\mathbf{z}|\mathbf{z}', \mathbf{x}) = T(\mathbf{z}', \mathbf{x})$. Specifically, we use an autoregressive scheme to progressively sample each code, *i.e.* $p(\mathbf{z}|\mathbf{z}', \mathbf{x}) = \prod_t p(\mathbf{z}_t|\mathbf{z}^{(k)}_{<t}, \mathbf{z}', \mathbf{x})$. At the t-th step, the transformer takes as input the current latent code $\mathbf{z}_{<t}$, the guidance code $\mathbf{z}'$ as well as encoded sensory feature $\mathbf{x}$ as input tokens, and outputs the next token's probability over the codebook $p(\mathbf{z}_t|\mathbf{z}^{(k)}_{<t}, \mathbf{z}', \mathbf{x})$:

$$\mathbf{z}^{(k)}_t \sim p(\mathbf{z}_t|\mathbf{z}^{(k)}_{<t}, \mathbf{z}', \mathbf{x}) \tag{2}$$

where $p(\mathbf{z}_t|\mathbf{z}^{(k)}_{<t}, \mathbf{z}', \mathbf{x})$ is the conditional probability estimated from the transformer. We use nucleus sampling [25] to get multiple diverse $\mathbf{z}^{(k)}$, which trades-off between sampling quality and diversity. Finally, these samples are decoded into multiple output samples using a decoder $G(\cdot)$: $\mathbf{y}^{(k)} = G(\mathbf{z}^{(k)})$, which provides our final layout estimation samples. Formally speaking, the entire second stage can be written as follows:

$$\mathbf{y}^{(k)} = G(\mathbf{z}^{(k)}) \;\text{ where }\; \mathbf{z}^{(k)} \sim p(\mathbf{z}|\mathbf{z}', \mathbf{x}). \tag{3}$$

One-step generation. To enhance inference speed, we introduce a one-step variant of MapPrior, which generates a single sample rather than multiple diverse ones. It also produces tokens in a single step, bypassing the autoregressive sampling strategy:

$$\mathbf{y} = G(\mathbf{z}) \;\text{ where }\; \mathbf{z} \sim p(\mathbf{z}|\mathbf{x}). \tag{4}$$

This provides an effective way to trade off between the generation quality and efficiency.

3.3. Learning

The training process consists of three individual components. Firstly, the perception module, denoted as F, is learned to produce a reliable initial layout estimation. Secondly, the encoder, decoder, and codebook (denoted as E, G, and $\mathcal{C}$, respectively) are jointly trained to represent a strong map prior model. Lastly, given a fixed map prior model, the conditional sampling transformer T is learned to sample high-quality final results. In the following, we will provide a detailed description of each component's training procedure.

Training the perception module. To train the perception module F, we follow the standard practice adopted in prior works [49, 56] and employ a binary cross-entropy loss. The objective function is defined as follows: $\min_F -[\mathbf{y}_{\text{gt}} \log \mathbf{y}']$, where $\mathbf{y}_{\text{gt}}$ denotes the ground-truth label, and $\mathbf{y}' = F(\mathbf{x})$ is the corresponding prediction given the input $\mathbf{x}$.

Training the map prior model. The training procedure for the map prior model involves jointly optimizing the encoder E, decoder G, and the fixed-size codebook $\mathcal{C}$ in an end-to-end fashion. We use a vector-quantized auto-encoder

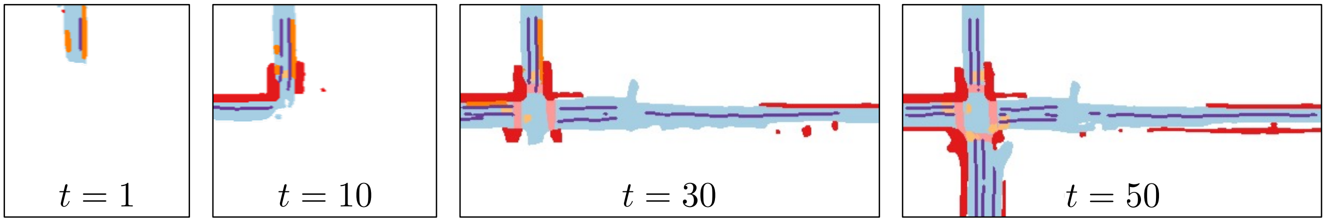

Figure 3: MapPrior can be exploited in a progressive manner to generate perpetual traffic layouts. At each step, we choose a local area to sample to fill unexplored regions and expand the frontiers.

to reconstruct the input data $\mathbf{y}$ through $\hat{\mathbf{y}} = G(\mathbf{z}(\mathbf{y}_i))$ following Eq. 1. The following losses are minimized:

$$\min_{E,G,\mathcal{C}} \max_{D} \mathcal{L}_{\text{recon}} + \lambda_{\text{GAN}}\mathcal{L}_{\text{GAN}} + \mathcal{L}_{\text{latent}}. \tag{5}$$

The first term, $\mathcal{L}_{\text{recon}}$, is the reconstruction loss, which is designed to maximize the agreement between the input $\mathbf{y}$ and the reconstructed output:

$$\mathcal{L}_{\text{recon}} = \|\hat{\mathbf{y}} - \mathbf{y}\|^2. \tag{6}$$

In addition to the reconstruction loss, the second term is a GAN loss inspired by VQGAN [18] to encourage the reconstructed output to be realistic with a clear local structure and topology. A local discriminator $D(\cdot)$ is trained to differentiate between real and reconstructed BEV layout using cross-entropy loss, and the $\mathcal{L}_{\text{GAN}}$ loss aims to make the reconstructed image as realistic as possible through fooling the discriminator:

$$\mathcal{L}_{\text{GAN}} = \log D(\mathbf{y}) + \log(1 - D(\hat{\mathbf{y}})), \tag{7}$$

where $\mathbf{y}$ is a real sample and $\hat{\mathbf{y}}$ is a reconstructed sample. Following VQGAN [18], this loss is rescaled by

$$\lambda_{\text{GAN}} = g_{\text{rec}}/(g_{\text{GAN}} + \sigma), \tag{8}$$

where g_{rec} and g_{GAN} are gradients of $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{GAN}}$ with respect to the last layer of the generator.

The last term promotes the expressiveness of the codebook. We expect the latent feature $E(\mathbf{y})$ to be close to its nearest codebook token and vice versa. Thus, the latent loss is defined as

$$\mathcal{L}_{\text{latent}} = \|\mathbf{z} - \text{sg}[E(\mathbf{y})]\|^2 + \|\text{sg}[\mathbf{z}] - E(\mathbf{y})\|^2, \tag{9}$$

where $\text{sg}[\cdot]$ is a gradient detach operator.

Training the conditional sampler. Our conditional sampling transformer, denoted as T, is trained in the latent code space to sample high-likelihood latent codes given input controlling guidance. Given a paired input and output $(\mathbf{x}, \mathbf{y})$ and fixed perception module F, map prior model $E, G, \mathcal{C}$, the transformer is trained using the following objective:

$$\min_{T} \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{out}}. \tag{10}$$

The first loss, $\mathcal{L}_{\text{CE}}$, optimizes the transformer to maximize the estimated probability of the latent code of the ground-truth layout. We use a cross-entropy loss between the transformer's estimated probability $T(\mathbf{z}', \mathbf{x})$ and the ground-truth sample $\mathbf{y}$:

$$\mathcal{L}_{\text{CE}} = \sum_t \sum_i \mathbf{y}_{i,t} \log T_{i,t}(\mathbf{z}', \mathbf{x}), \tag{11}$$

where, i represents the i-th output label, and t represents the t-th autoregressive step.

Although making the latent code closer to the ground-truth map's latent code is essential, it is not sufficient to ensure high output fidelity. Hence, we include an additional reconstruction output loss that encourages the transformer to favor samples that produce high-accuracy layouts. Similar to the reconstruction loss in auto-encoding training, an L2 reconstruction loss is used:

$$\mathcal{L}_{\text{out}} = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2, \tag{12}$$

where $\mathbf{y}$ is the true map and $\hat{\mathbf{y}}$ is the predicted map. Note that codebook selection in latent code space is non-differentiable; thus, we use Gumbel-Softmax [29] to ensure differentiability in practice throughout the training processes.

3.4. Discussions

Uncertainty quantification. Our latent transformer offers diverse results $\mathbf{y}^{(k)}$ by using a conditional sampling scheme. By estimating the variance of $\mathbf{y}^{(k)}$, we can estimate the uncertainty map for our results. By using the average of $\mathbf{y}^{(k)}$, we aggregate the diverse result samples to a more stable and better calibrated results.

Perpetual layout generation. Our trained map prior enables the continuous generation of realistic and varied driving sequences, which are extremely valuable for content creation and autonomous driving simulations. We use a progressive generation strategy, which builds upon prior works [3, 17, 45, 65, 18, 9]. The strategy involves iteratively expanding our visual horizon and generating new content by leveraging our map prior model to fill in the previously unseen areas. We illustrate this process in Fig. 3.

Table 1: Quantitative results of BEV map segmentation on nuScenes. Our MapPrior achieves better accuracy (IoU), realism (MMD) and uncertainty awareness (ECE) than discriminative BEV perception baselines.

	Modality	IoU (↑)							MMD (↓)	ECE (↓)
		Drivable	Ped. X	Walkway	Stop Line	Carpark	Divider	Mean		
OFT [60]	C	74.0	35.3	45.9	27.5	35.9	33.9	42.1	54.5	0.045
LSS [56]	C	75.4	38.8	46.3	30.3	39.1	36.5	44.4	43.2	0.041
BEVFusion-C [49]	C	**81.7**	**54.8**	**58.4**	**47.4**	50.7	**46.4**	56.6	39.6	0.038
MapPrior-C	C	**81.7**	54.6	58.3	46.7	**53.3**	45.1	**56.7**	**28.4**	**0.026**
MapPrior-C (1 step)	C	81.6	54.6	**58.4**	46.8	**53.9**	45.1	56.7	28.7	–
PointPillars [38]	L	72.0	43.1	53.1	29.7	27.7	37.5	43.8	153.0	0.111
BEVFusion-L [80, 49]	L	75.6	48.4	57.5	36.5	31.7	41.9	48.6	115.2	0.090
MapPrior-L	L	**81.0**	**49.7**	**58.0**	**37.5**	**38.2**	**42.4**	**51.1**	**35.8**	**0.038**
MapPrior-L (1 step)	L	80.1	49.0	57.8	37.8	33.0	42.5	50.0	50.2	–
PointPainting [75]	C+L	75.9	48.5	57.1	36.9	24.5	41.9	49.1	109.8	0.099
MVP [81]	C+L	76.1	48.7	57.0	36.9	33.0	42.2	49.0	115.3	0.096
BEVFusion [49]	C+L	85.5	60.5	67.6	52.0	57.0	53.7	62.7	**21.6**	0.038
MapPrior-CL	C+L	85.3	**61.4**	67.1	**51.7**	**60.0**	53.3	**63.1**	28.0	**0.020**
MapPrior-CL (1 step)	C+L	85.3	61.3	67.0	51.7	59.6	53.1	63.0	28.1	–

4. Experiments

We evaluate our MapPrior on BEV map segmentation and generation tasks for both LiDAR and camera modalities. We evaluate our approach to generate an accurate and realistic traffic layout both quantitatively and qualitatively in Sec. 4.2. We show how output loss and BEV features can affect our performance in Tab. 2. We finally estimate how our model can generate diverse samples and how our model is calibrated using the diverse samples in Fig. 5 and 6. Specifically, we are interested in seeing how using a generative prior affects accuracy (reflected by mIoU), realism (reflected by MMD), and model calibration (reflected by ECE).

4.1. Experimental Setup

Datasets. We evaluate our model on nuScenes [6], a large-scale outdoor autonomous driving dataset containing 1000 driving scenes, consisting of 700 scenes for training, 150 for validation, and 150 scenes for testing. It has around 40,000 annotated key-frames, each with six monocular camera images encompassing a 360-degree FoV (used by camera models), and a 32-beam LiDAR scan (used by LiDAR models). We follow the train/validation split provided by nuScenes. We evaluate all models on the validation set following common practices in BEV layout estimation [49, 56]. The ground truth segmentation map is provided by nuScenes and was labeled manually [6]. We rasterized the map layers from the nuScenes map into the ego frame.

Metrics. We provide quantitative results for both segmentation and generative tasks. For BEV map segmentation, our metric is the Intersection-over-Union (IoU score) for six map classes (drivable area, pedestrian crossing, walkway, stop line, car-parking area, and lane divider), as well as the mean IoU averaged over all six map classes. Since different classes may overlap, we separately compare the IoU score for each class. For all baseline predictive models, we choose the best threshold that maximizes IoU for comparison to ensure no bias is introduced due to the suboptimal threshold selection. For MapPrior, we simply use 0.5 as the threshold.

To evaluate generated map layout realism, we use the maximum mean discrepancy (MMD) metric. MMD effectively measures the distance between two distributions of two sets of samples by measuring the squared difference of their mean in different spaces:

$$\text{MMD} = \sum_{i}^{n}\sum_{i'}^{n} k(\mathbf{x}_i, \mathbf{x}_{i'})/n^2 + \sum_{j}^{m}\sum_{j'}^{m} k(\mathbf{x}_j, \mathbf{x}_{j'})/m^2 \\ -2 \cdot \sum_{i}^{n}\sum_{j}^{m} k(\mathbf{x}_i, \mathbf{x}_j)/nm. \quad (13)$$

To evaluate a set of predicted layouts, we compare it with a set of ground truth layouts from nuScenes.

To evaluate uncertainty in our diversity results, we use Expected Calibration Error (ECE) [36, 22]. ECE compares output probabilities to model accuracy. It splits the results into several bins and measures a weighted average discrepancy between accuracy and confidence within each bin:

$$\text{ECE} = \sum_{b}^{B} n_b |\mathsf{acc}(b) - \mathsf{conf}(b)|/n, \quad (14)$$

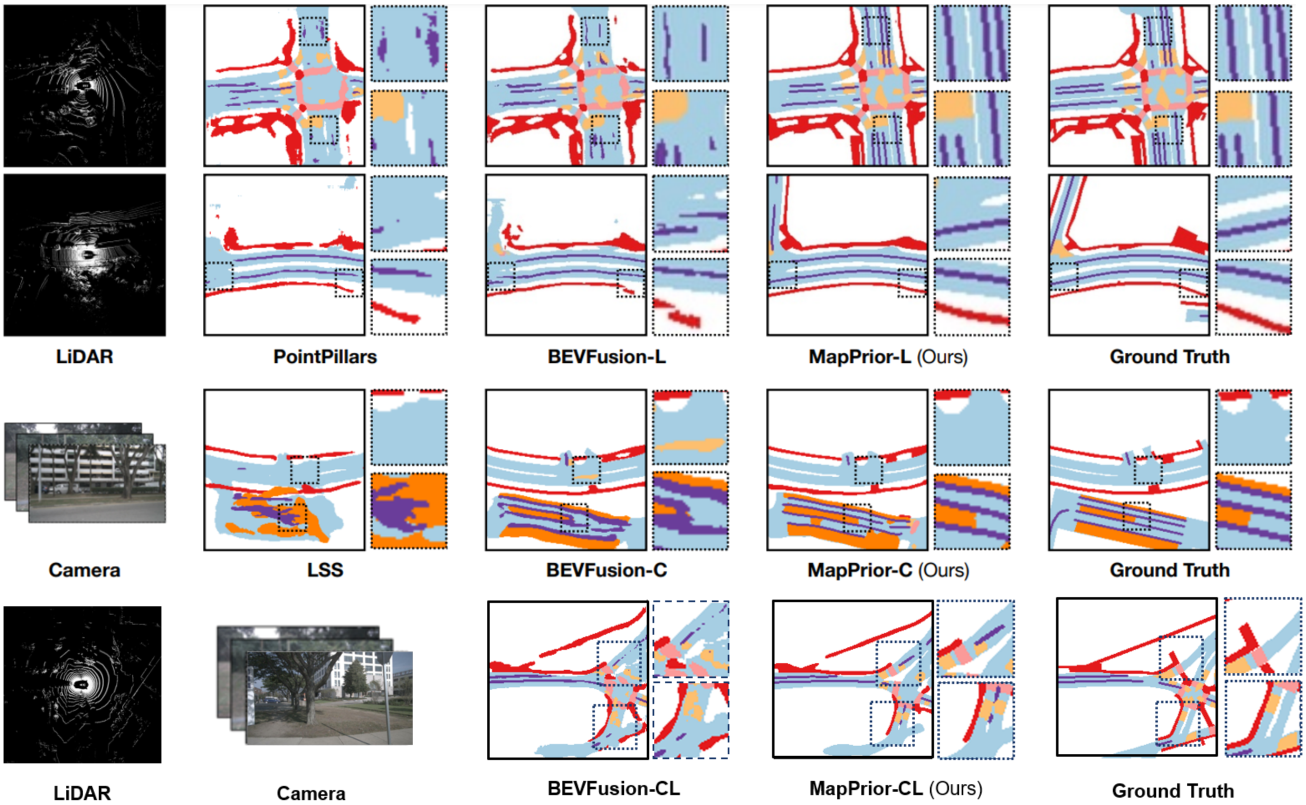

Figure 4: Qualitative results of BEV map segmentation on nuScenes. Discriminative BEV perception baselines produce results with clear artifacts (*e.g.*, lane markings are discontinuous, roads and pedestrian walkways have unrealistic gaps, *etc*.). In contrast, our MapPrior produces semantic map layouts that are clean, realistic, and in-distribution.

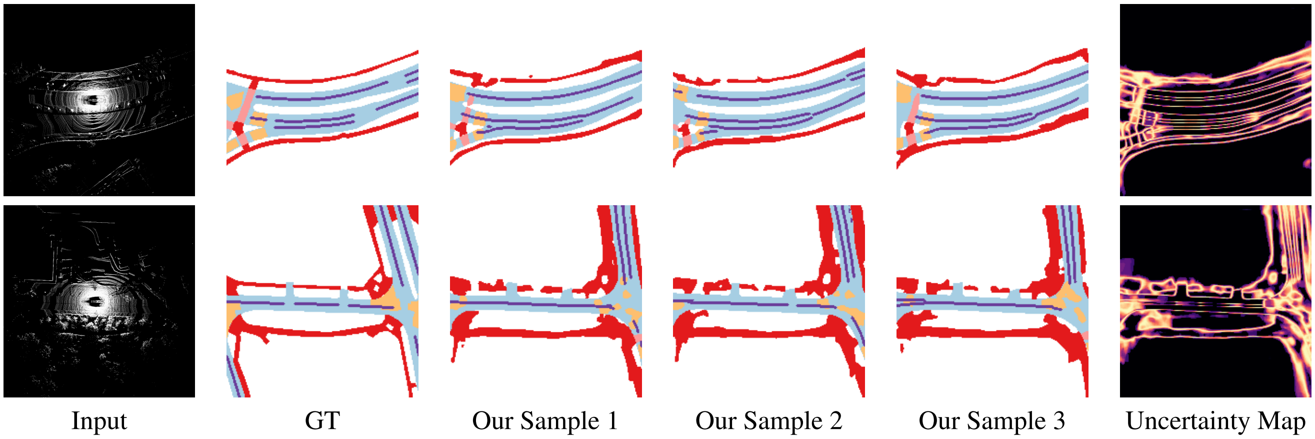

Figure 5: Qualitative results of diversity on NuScenes.

where $\texttt{acc}$ is the empirical accuracy, and $\texttt{conf}$ is the estimated confidence for each bin, n_b is the # of samples per each bin and n is the total # of samples. ECE is an important metric in BEV map segmentation, as ignoring uncertainties can lead to bad consequences in driving planning. We generate 15 diverse outputs for every instance and use the mean as the confidence. The baseline model is trained end-to-end with cross-entropy loss, so the predictions produced by the softmax function are assumed to be the pseudo probabilities.

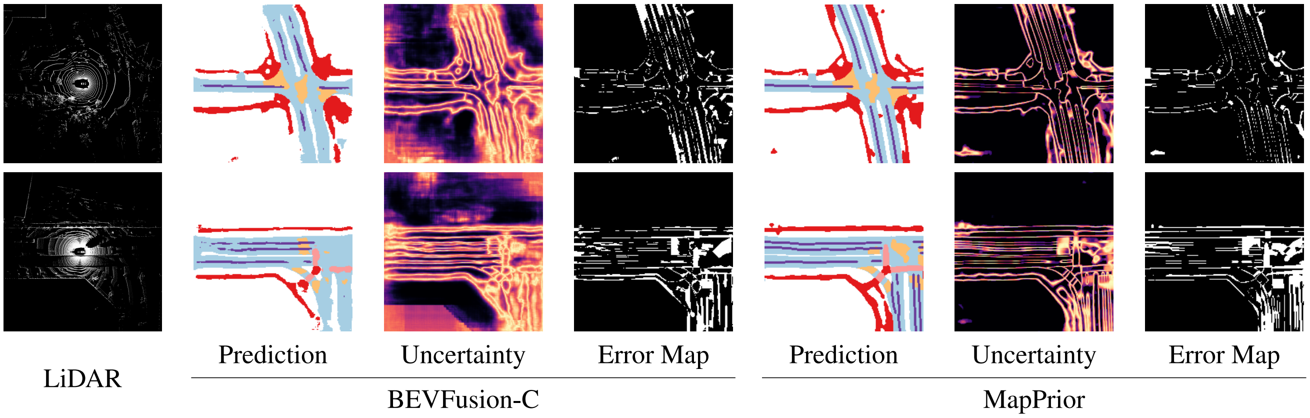

Figure 6: Prediction, uncertainty, and error map comparison between BEVFusion-C and MapPrior. BEVFusion generates a misguided uncertainty boundary by giving high weight to areas where map elements are unlikely to be. In comparison, MapPrior's uncertainty is constrained to map element boundaries. This is further confirmed with the error map, where MapPrior's uncertainty map closely matches the error map.

Table 2: Ablation studies for MapPrior on nuScenes.

Output Loss $\mathcal{L}_{\text{out}}$	$T(\mathbf{z}')$ or $T(\mathbf{z}', \mathbf{x})$	mIoU (↑)	MMD (↓)
–	$T(\mathbf{z}')$	45.6	**33.3**
✓	$T(\mathbf{z}')$	49.0	45.6
✓	$T(\mathbf{z}', \mathbf{x})$	**51.1**	35.8

Baselines. For LiDAR-only segmentation, we use PointPillars [38], and BEVFusion-L [49, 80] as our baselines. To the best of our knowledge, BEVFusion-L is the current state-of-the-art method in LiDAR-based BEV map segmentation.

For camera-only segmentation, our baseline model are LSS [56], OFT [60], and BEVFusion-C [49]. Among them, BEVFusion-C is the current state-of-the-art model with a significantly higher IoU score than other methods. We use open-source code from BEVFusion. We also provide results of multi-modal models [49, 75, 81] for reference.

Implementation details. Following [49], we perform segmentation in a [-50m, 50m]×[-50m, 50m] region around the ego car with a resolution of 0.5 meters/pixel, resulting in a final image size of 200 × 200. Since map classes may overlap, our model performs binary segmentation for all classes. The encoder and decoder comprise four downsampling and four upsampling blocks processing a series of 128-256-512-256-128 channels. Each block comprises 2 resnet blocks and one convolution and uses sigmoid activation and GroupNorm. The resolution of the latent space is [12,12].

For the generative step, we train a minGPT [30] transformer conditioned on the generated sequence, extracted BEV features, and the initial noisy map. We train the whole model using Adam [32] with a learning rate of 9.0e-6. The transformer's BEV feature encoder for the transformer has a similar structure to the model encoder consisting of 3 downsampling blocks. The BEV feature encoder converts the original BEV features shaped [128, 128] into latent space tokens shaped [12, 12]. We apply a multiplier of 100 on the output loss to balance the magnitude of different losses.

4.2. Map Segmentation as Generation

Quantitative results. We show our quantitative results for map segmentation in Tab. 1. The results show that MapPrior achieves state-of-the-art performance. Comparing with BEVFusion-L, our MapPrior-L offers **2.5%** improvement in mIoU, which is brought purely by our proposed generative stage. Furthermore, MapPrior provides a substantial improvement in the MMD score compared to baselines. MMD is a metric of distance between the generated layout predictions and the ground truth distribution. This shows that MapPrior's outputs closely match the ground truth data distribution. In addition, this stark difference in MapPrior's MMD performance compared to the baselines implies that the realism metric and precision metric are not closely coupled. It is possible to achieve higher IoU while generating non-realistic samples, or vice versa. Our approach simultaneously pushes the limit of the two.

Moreover, inference speed is vital for MapPrior. Using an RTX A6000 GPU, we compared the inference speeds of our model and BEVFusion in terms of frames per second (FPS). Our findings are presented in Tab. 3. These results indicate that one-step MapPrior is significantly closer to real-time performance compared to the standard MapPrior, with only a minor trade-off in IoU for increased uncertainty awareness.

Qualitative results. We show our qualitative results in Fig. 4. Compared to other methods, MapPrior can consis-

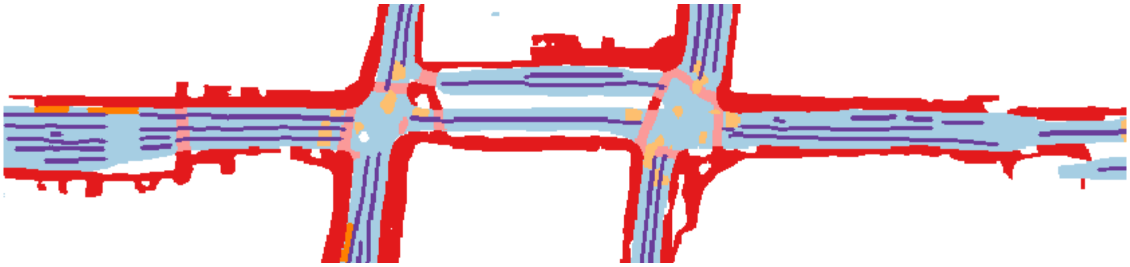

Figure 7: Results for perpetual traffic scene generation. Two road subsections are shown here.

Table 3: Performance of one-step MapPrior.

	Modality	mIoU (↑)	MMD (↓)	FPS
BEVFusion-C [49]	C	56.6	39.6	8.85
MapPrior-C	C	**56.7**	**28.4**	0.60
MapPrior-C (1 step)	C	56.7	28.7	4.26
BEVFusion-L [80, 49]	L	48.6	115.2	7.52
MapPrior-L	L	**51.1**	**35.8**	0.57
MapPrior-L (1 step)	L	50.0	50.2	4.88
BEVFusion [49]	C+L	62.7	**21.6**	5.52
MapPrior-CL	C+L	**63.1**	28.0	0.55
MapPrior-CL (1 step)	C+L	63.0	28.1	3.61

tently generate coherent and realistic class predictions within the entire map region. Our model has a more coherent divider layout, whereas the baseline methods usually have broken/missing lane dividers. In the baseline methods, the stopline is often jagged and appears disconnected from the road. In distant areas from the ego car, our model tries to make a plausible layout estimate when other methods fail due to limited observations (this is especially noticeable in the pedestrian walkways). The edges from our method are also more smooth, resulting in a more realistic estimation.

Diversity and uncertainty calibration. We show that our model can produce a better-calibrated layout with diversity. In Fig. 5, we show that our model can generate multiple diverse and feasible layouts, All of which are realistic.

By aggregating the diverse samples, our model can produce a calibrated uncertainty map. We show our results in Fig. 6. Compared to the baselines (which are unable to generate multiple samples), our uncertainty map aligns with the error map much more accurately. ECE scores in Tab. 1 further validate this quantitatively.

Perpetual generation. We show our qualitative results for generating 'infinite' roads in Fig. 7. We have generated a single 30km long road. Due to size constraints, we are only able to show a subsection. The generated traffic scene contains a highway with intermittent intersections resembling a road layout in a city. We provide the entire road as a gif in our supplementary materials.

4.3. Discussions

Ablation studies. To justify our design, we provide ablation studies in Tab. 2. L2 loss at the output end changes the model's optimization target. It puts more weight on generating accurate results as opposed to generating i.i.d. data. Therefore, the model achieves a higher IoU score at the cost of a slightly worse MMD. Providing the transformer with BEV features boosts the performance by around 3% in the IoU score, and decreases MMD by around 21%.

Inference speed. Inference speed is essential for MapPrior, especially given the real-time demands of autonomous driving. While transformers offer remarkable generative abilities, they inherently slow down performance due to their sequential token generation process. Moreover, the need for MapPrior to produce diverse outcomes further curtails its inference speed. To address this, we introduced the one-step variant of MapPrior, which predicts a single sample for each input and generates all tokens simultaneously. As evidenced in Tab. 3, the one-step MapPrior registers a marginally worse MMD score and falls short in uncertainty awareness. Nonetheless, the one-step MapPrior markedly outpaces the standard MapPrior in speed.

5. Conclusion

This paper presents MapPrior, a novel generative method for performing BEV perception. The core idea is to leverage a learned generative prior over traffic layouts to provide diverse and accurate layout estimations, which potentially enable more informed decision-making and motion planning. Our experiments show that our approach produces more realistic scene layouts, enhances accuracy, and better calibrates uncertainty compared to current methods.

Acknowledgement. The project is partially funded by the Illinois Smart Transportation Initiative STII-21-07, an Amazon Research Award, an Intel Research Gift, and an IBM IIDAI grant. We also thank NVIDIA for the Academic Hardware Grant. VZ is supported by the Frontier Fellowship.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017. 2

[2] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *CVPR*, 2022. 2

[3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. 2009. 5

[4] Ioan Andrei Barsan, Shenlong Wang, Andrei Pokrovsky, and Raquel Urtasun. Learning to localize using a lidar intensity map. In *CoRL*, 2018. 2

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019. 2

[6] 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 *CVPR*, 2020. 6

[7] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *CoRL*, 2018. 2

[8] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. In *CVPR*, 2021. 2

[9] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022. 2, 3, 5

[10] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 2

[11] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. Neat: Neural attention fields for end-to-end autonomous driving. In *CVPR*, 2021. 2

[12] Min Jin Chong and David Forsyth. Jojogan: One shot face stylization. In *ECCV*, 2022. 3

[13] Laurène Claussmann, Marc Revilloud, Dominique Gruyer, and Sébastien Glaser. A review of motion planning for highway autonomous driving. 2020. 2

[14] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 2

[15] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. In *ICLR (Workshop)*, 2014. 2

[16] Shivam Duggal, Shenlong Wang, Wei-Chiu Ma, Rui Hu, and Raquel Urtasun. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *ICCV*, 2019. 3

[17] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, 2001. 5

[18] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2, 3, 4, 5

[19] W.T. Freeman and E.C. Pasztor. Learning low-level vision. In *CVPR*, 1999. 3

[20] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NeurIPS*, 2014. 2

[21] Nikhil Gosala and Abhinav Valada. Bird's-eye-view panoptic segmentation using monocular frontal view images. 2022. 2

[22] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 6

[23] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, 2018. 2

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2

[25] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *ICLR*, 2020. 4

[26] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. FIERY: Future instance segmentation in bird's-eye view from surround monocular cameras. In *ICCV*, 2021. 2

[27] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. BEVDet: High-Performance Multi-Camera 3D Object Detection in Bird-Eye-View. *arXiv preprint arXiv:2112.11790*, 2021. 2

[28] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2, 3

[29] Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. In *ICLR*, 2017. 5

[30] Andrej Karpathy. Mingpt. https://github.com/karpathy/minGPT, 2022. 8

[31] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 2

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 8

[33] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018. 2

[34] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2

[35] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011. 3

[36] Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration. In *NeurIPS*, 2019. 6

[37] Markus Kängsepp and Meelis Kull. Calibrated perception uncertainty across objects and regions in bird's-eye-view. In *NeurIPS (Workshop)*, 2022. 2

[38] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 2, 6, 8

[39] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J. Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, Michael Sokolsky, Ganymed Stanek, David Stavens, Alex Teichman, Moritz Werling, and

Sebastian Thrun. Towards fully autonomous driving: Systems and algorithms. In *IV*, 2011. 2

[40] Jesse Levinson and Sebastian Thrun. Robust vehicle localization in urban environments using probabilistic maps. In *ICRA*, 2010. 2

[41] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *CVPR*, 2021. 3

[42] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *ICRA*, 2022. 2

[43] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, 2020. 2

[44] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *ECCV*, 2018. 2

[45] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *ICCV*, 2021. 5

[46] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *ECCV*, 2022. 2

[47] Yicheng Liu, Yuan Yuantian, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. *arXiv preprint arXiv:2206.08920*, 2022. 2

[48] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *ICCV*, 2021. 4

[49] 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 *ICRA*, 2023. 1, 2, 3, 4, 6, 8, 9

[50] W. Luo, B. Yang, and R. Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *CVPR*, 2018. 2

[51] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. In *CVPR*, 2021. 2

[52] Gellert Mattyus, Shenlong Wang, Sanja Fidler, and Raquel Urtasun. Hd maps: Fine-grained road segmentation by parsing ground and aerial images. In *CVPR*, 2016. 2

[53] Andrew Ng and Michael Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NeurIPS*, 2001. 2

[54] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. In *NeurIPS*, 2020. 3

[55] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, 2018. 2

[56] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020. 1, 2, 3, 4, 6, 8

[57] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *CVPR*, 2021. 2

[58] Marc'Aurelio Ranzato, Joshua Susskind, Volodymyr Mnih, and Geoffrey Hinton. On deep generative models with applications to recognition. In *CVPR*, 2011. 2

[59] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, 2021. 2

[60] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. In *BMVC*, 2019. 2, 6, 8

[61] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[62] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2

[63] S. Roth and M.J. Black. Fields of experts: a framework for learning image priors. In *CVPR*, 2005. 3

[64] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, 2020. 2

[65] Yuan Shen, Wei-Chiu Ma, and Shenlong Wang. SGAM: Building a virtual 3d world through simultaneous generation and mapping. In *NeurIPS*, 2022. 5

[66] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 3

[67] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. Springer, 2023. 2

[68] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019. 2

[69] Nitish Srivastava and Russ R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NeurIPS*, 2012. 2

[70] Deqing Sun, Stefan Roth, and Michael J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. 3

[71] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. *Stanley: The Robot That Won the DARPA Grand Challenge*. 2007. 2

[72] Hung-Yu Tseng, Lu Jiang, Ce Liu, Ming-Hsuan Yang, and Weilong Yang. Regularizing generative adversarial networks under limited data. In *CVPR*, 2021. 2

[73] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. In *NeurIPS*, 2020. 2

[74] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017. 2

[75] Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *CVPR*, 2020. 2, 6, 8

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

[77] Koichiro Yamaguchi, Tamir Hazan, David McAllester, and Raquel Urtasun. Continuous markov random fields for robust stereo estimation. In *ECCV*, 2012. 3

[78] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10), 2018. 4

[79] R. A. Yeh, C. Chen, T. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *CVPR*, 2017. 2

[80] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. In *CVPR*, 2021. 1, 2, 3, 4, 6, 8, 9

[81] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Multimodal virtual point 3d detection. In *NeurIPS*, 2021. 2, 3, 6, 8

[82] Wenyuan Zeng, Shenlong Wang, Renjie Liao, Yun Chen, Bin Yang, and Raquel Urtasun. Dsdnet: Deep structured self-driving network. In *ECCV*, 2020. 2

[83] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019. 2

[84] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, Congcong Li, and Dragomir Anguelov. Tnt: Target-driven trajectory prediction. In *CoRL*, 2020. 2

[85] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2

[86] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 3

[87] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *CVPR*, 2022. 2

[88] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 3

[89] Song Chun Zhu, Yingnian Wu, and D. Mumford. Frame: filters, random fields, and minimax entropy towards a unified theory for texture modeling. In *CVPR*, 1996. 3