Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

Self-supervised Multi-future Occupancy Forecasting for Autonomous Driving

Bernard Lange*, Masha Itkina*, Jiachen Li† and Mykel J. Kochenderfer*
*Stanford University, †University of California, Riverside

***Abstract*—Environment prediction frameworks are critical for the safe navigation of autonomous vehicles (AVs) in dynamic settings. LiDAR-generated occupancy grid maps (L-OGMs) offer a robust bird's-eye view for the scene representation, enabling self-supervised joint scene predictions while exhibiting resilience to partial observability and perception detection failures. Prior approaches have focused on deterministic L-OGM prediction architectures within the grid cell space. While these methods have seen some success, they frequently produce unrealistic predictions and fail to capture the stochastic nature of the environment. Additionally, they do not effectively integrate additional sensor modalities present in AVs. Our proposed framework, Latent Occupancy Prediction (LOPR), performs stochastic L-OGM prediction in the latent space of a generative architecture and allows for conditioning on RGB cameras, maps, and planned trajectories. We decode predictions using either a single-step decoder, which provides high-quality predictions in real-time, or a diffusion-based batch decoder, which can further refine the decoded frames to address temporal consistency issues and reduce compression losses. Our experiments on the nuScenes and Waymo Open datasets show that all variants of our approach qualitatively and quantitatively outperform prior approaches.**

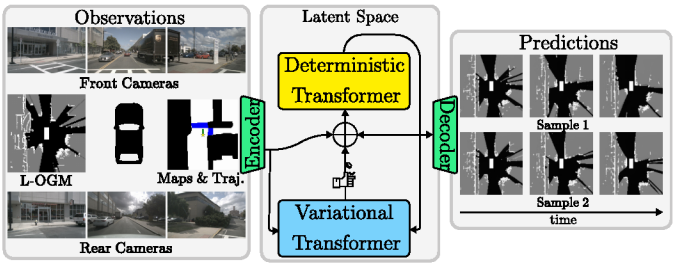

Fig. 1: Latent Occupancy PRediction (LOPR) is a self-supervised stochastic prediction framework that forecasts occupancy grid maps within the latent space of a generative model. It consists of deterministic and variational transformer modules conditioned on occupancy grids, images, maps, and the planned trajectory. LOPR forecasts multiple plausible futures for the entire scene.

I. Introduction

Accurate environment prediction algorithms are essential for autonomous vehicle (AV) navigation in urban settings. Experienced drivers understand scene semantics and recognize the intent of other agents to anticipate their trajectories and safely navigate to their destination. To replicate this process in AVs, many environment prediction approaches have been proposed, employing different environment representations and modeling assumptions [21, 26, 42, 4, 6, 31, 27].

The modern AV stack comprises a mixture of expert-designed and learned modules, such as 3D object detection, tracking, motion forecasting, and planning, each developed independently. In the case of learned systems, development involves using curated labels provided by human annotators and other perception systems. For environmental reasoning, object-based prediction algorithms are often used, which rely on the perception system to create a vectorized representation of the scene with defined agents and environmental features [4, 33]. However, this approach has multiple limitations. First, it often generates marginalized future trajectories for each individual agent, rather than a holistic scene prediction including agent interactions, which complicates integration with planning modules [5]. Second, this approach does not take sensor measurements into account and depends solely on object detection algorithms that may fail in suboptimal conditions [7, 10]. Third, reliance on labeled data, sourced from both human annotators and perception systems, constrains the dataset size and incurs higher costs. These drawbacks render the AV stack susceptible to cascading failures and can lead to poor generalization in unforeseen scenarios. Such limitations underscore the need for complementary environment modeling approaches that do not rely on error-prone and expensive labeling schemes.

In response to these challenges, occupancy grid maps generated from LiDAR measurements (L-OGMs) have gained popularity as a form of scene representation for prediction. The popularity is due to their minimal data preprocessing requirements, eliminating the need for manual labeling, the ability to model the joint prediction of the scene with an arbitrary number of agents, and robustness to partial observability and detection failures [21, 26]. We focus on ego-centric L-OGM prediction generated using uncertainty-aware occupancy state estimation approaches [11]. Due to its generality and ability to scale with unlabeled data, we hypothesize that L-OGM prediction alongside RGB video prediction could also serve as an unsupervised pre-training objective, i.e., a foundational model, for autonomous driving.

The task of L-OGM prediction is typically framed as self-supervised sequence-to-sequence learning. ConvLSTM-based architectures have been used predominantly in previous work for this task due to their ability to handle spatiotemporal sequences [21, 37, 26, 42]. These approaches are optimized end-to-end in grid cell space, do not account for the stochasticity present in the scene, and neglect other available modalities, e.g., RGB cameras around the vehicle, maps, and the planned

trajectory. As a result, they often suffer from unrealistic and blurry predictions.

In this work, we address the limitations of previous approaches by proposing a stochastic L-OGM prediction framework that operates within the latent space of a generative model [28]. Generative models are known for providing a compressed representation, while producing high-quality samples [14, 25]. With the use of generative models, we can minimize redundancies in the representation, allowing the prediction network to focus computation on the most critical aspects of the task [50]. Vector-quantized variational models [45] combined with autoregressive transformers [46] have demonstrated significant success. However, it often comes at the expense of increased inference time, driven by the large number of discrete tokens needed to effectively capture the task [36]. In this work, we leverage lower dimensional continuous representations to enable real-time performance.

Within the latent space trained on L-OGMs, our framework employs an autoregressive transformer-based architecture consisting of two modules, called sequentially at each time step: a variational module that models the stochasticity of the scene and a deterministic module that predicts the next time step. Both are conditioned on past L-OGM encodings and other modalities if available, such as camera images, maps, and the planned trajectory, as shown in Fig. 1. Predictions are decoded one by one using a single-step decoder, which provides high-quality predictions in real-time that optionally can be refined with a diffusion-based batch decoder. The diffusion-based batch decoder addresses the temporal consistency issues associated with single-step decoders [19] and mitigates compression losses by conditioning on prior rasterized L-OGMs, at the cost of real-time feasibility.

Experiments on nuScenes [3] and the Waymo Open Dataset [40] show quantitative and qualitative improvements over prior approaches, including recent state-of-the-art discrete transformer-based approaches. Our framework forecasts diverse futures and infers unobserved agents. It also leverages other sensor modalities for more accurate predictions, such as observing oncoming vehicles in a camera feed beyond the visible region of the L-OGMs. Our contributions are:

- We introduce a framework called **L**atent **O**ccupancy **PR**ediction (LOPR) for stochastic L-OGM prediction in the latent space of a generative model conditioned on other sensor modalities, such as RGB cameras, maps, and the planned AV trajectory.
- We propose a variational-based transformer model that captures the stochasticity of the surrounding scene while remaining real-time feasible.
- We define a diffusion-based batch decoder that refines single-frame decoder outputs to address temporal consistency issues and reduce compression losses.
- Through experiments on the nuScenes [3] and Waymo Open Dataset [40], we demonstrate that LOPR surpasses prior L-OGM prediction methods and highlight the positive impact of incorporating additional input modalities.

II. Related Work

OGM Prediction. The majority of prior work in OGM prediction generates OGMs with LiDAR measurements (L-OGMs) and uses an adaptation on the recurrent neural network (RNN) with convolutions [43, 44]. Dequaire et al. [8] tracked objects through occlusions and predicted future binary OGMs with an RNN and a spatial transformer. Schreiber et al. [37] provided dynamic occupancy grid maps (DOGMas) with cell-wise velocity estimates as input to a ConvLSTM for environment prediction from a stationary platform. Schreiber et al. [38] then extended this work to forecast DOGMas in a moving ego-vehicle setting. Mohajerin and Rohani [32] applied a difference learning approach to predict OGMs as seen from the coordinate frame of the first observed time step. Itkina et al. [21] used the PredNet ConvLSTM architecture [30] to achieve ego-centric OGM prediction. Lange et al. [26] reduced the blurring and the gradual disappearance of dynamic obstacles in the predicted grids by developing an attention augmented ConvLSTM mechanism. Concurrently, Toyungyernsub et al. [42] addressed obstacle disappearance with a double-prong framework assuming knowledge of the static and dynamic obstacles. An alternative approach predicts occupancy grid maps from vectorized object data [31] or a mixture of vectorized object data and sensor measurements [52]. Similar to representations in common trajectory prediction techniques, these methods require substantial labeling efforts [12, 35, 41]. Unlike prior work, we perform self-supervised multi-future L-OGM predictions in the latent space of generative models conditioned on additional sensor modalities without the need for manual labeling.

Representation Learning in Robotics and Autonomous Driving. The objective of representation learning is to identify low-dimensional representations that make it easier to achieve the desired performance on a task. Many robotics applications use architectures such as the autoencoder (AE) [17], the variational autoencoder (VAE) [25], the generative adversarial network (GAN) [14], and the vector quantized variational autoencoder (VQ-VAE) [45]. Latent spaces have been used to learn latent dynamics from pixels [15], output video predictions [1], generate trajectories [16], and learn autonomous driving neural simulators [24, 19]. Large-scale video prediction architectures have used discrete representations provided by the VQ-VAE [45] with a causal transformer [46, 36, 19]. However, these models remain prohibitively expensive to train and sample from due to large number of discrete tokens required. We present a method that performs multi-future L-OGM prediction entirely in the continuous latent space of VAE-GAN generative model in real time.

III. LOPR: Latent Occupancy PRediction

We propose the **Latent Occupancy Prediction (LOPR)** framework, designed to generate stochastic scene predictions represented as ego-centric L-OGMs. A 2D L-OGM grid $x \in \mathbb{R}^{H \times W}$ represents a bird's-eye-view map with dimensions (H, W), where each cell encodes the probability of occupancy. The grid is generated using projected LiDAR sensor measurements

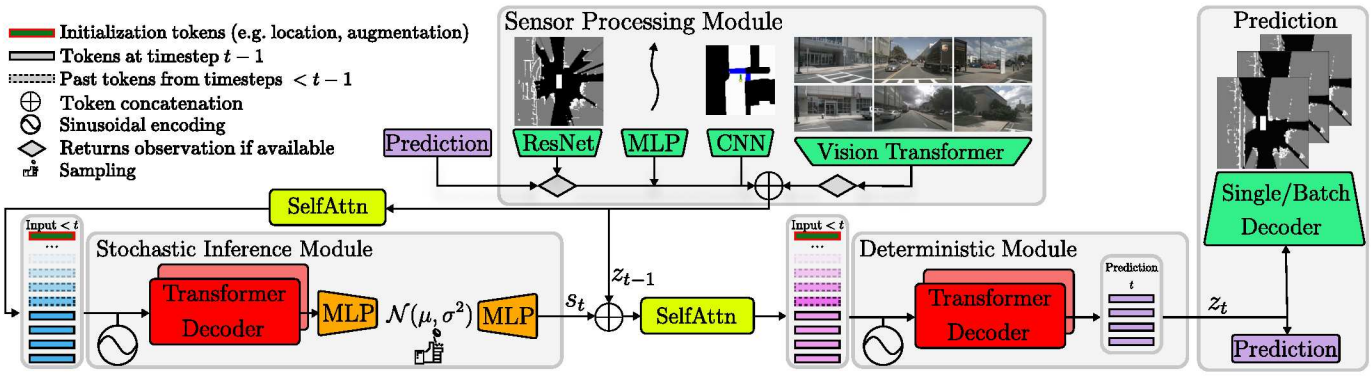

Fig. 2: The LOPR framework, comprising sensor processing, stochastic inference, and prediction modules. The sensor processing module encodes all sensor modalities. The L-OGM and RGB camera encoders are pretrained as described in Section III-A and in Section III-D. The inference module captures the scene's stochasticity (Section III-B). In the prediction module, we forecast the next time step's L-OGM embedding. At each time step, the most recent predictions are autoregressively provided to the inference and prediction modules.

with segmented ground. The task of stochastic prediction involves learning a distribution over future occupancy grids conditioned on the observed grids, $p(x_{>T_O} \mid x_{\leq T_O})$, where T_O denotes the observation horizon.

LOPR separates the task into (1) learning an L-OGM representation and (2) making predictions in the latent space of a generative model. In the representation learning phase, a VAE-GAN is trained to learn an L-OGM latent space. During the prediction stage, our framework uses an autoregressive transformer-based architecture, comprising both deterministic and variational decoder models. At each time step, a sample is drawn from the variational transformer and then passed to the deterministic transformer to forecast the next L-OGM embedding. Predictions are conditioned on past L-OGMs encodings and other available modalities, such as camera images, maps, and the planned trajectory. The encoders for maps and planned trajectories are trained alongside the prediction framework, while for the image encoder, we use a pre-trained DINOv2-based model [34]. Predictions are decoded using a single-step decoder, which provides high-quality predictions in real-time that optionally can be refined with a diffusion-based batch decoder. Fig. 2 summarizes the framework.

A. Representation Learning

The latent space for L-OGMs is obtained by training an encoder and a decoder. Given an L-OGM grid x, the encoder outputs a low-dimensional latent representation $z \in \mathbb{R}^{c \times h \times w}$ with dimensions (h, w) and depth c. This representation is reconstructed by a decoder to $\hat{x} \in \mathbb{R}^{H \times W}$. The framework integrates concepts from β-VAE and GAN [28, 24]. In β-VAE, the objective consists of the reconstruction and regularization losses:

$$\mathcal{L}_{\text{VAE}} = -\mathbb{E}_{q(z|x)} \left[\log p\left(x \mid z\right)\right] + \beta D_{\text{KL}}\left(q\left(z \mid x\right) || \, p\left(z\right)\right), \tag{1}$$

where $q\left(z \mid x\right)$ and $p\left(x \mid z\right)$ are the outputs of the encoder and decoder respectively, $p\left(z\right)$ is the unit Gaussian prior, and D_{KL} represents the Kullback-Leibler divergence. The reconstruction loss is an average of the perceptual loss [51] and the mean squared error. In the GAN step, the same decoder serves as the generator, and a discriminator classifies whether samples originate from the training set. This framework uses minimax optimization for the following objective [14]:

$$\mathcal{L}_{\text{GAN}} = \mathbb{E}_{x \sim p_{\text{data}}} \left[\log \mathcal{D}(x)\right] + \mathbb{E}_{\hat{x} \sim p_{\text{model}}} \left[\log\left(1 - \mathcal{D}(\hat{x})\right)\right], \tag{2}$$

where $\mathcal{D}$ is the discriminator. The final loss is $\mathcal{L}_{\text{rep}} = \mathcal{L}_{\text{VAE}} + \mathcal{L}_{\text{GAN}}$ and follows the implementation described by Karras et al. [23] and Kim et al. [24].

B. Stochastic L-OGM Sequence Prediction

Given the pre-trained L-OGM latent space, we train a stochastic sequence prediction network that receives a history of observations and outputs a distribution over a potential future embedding $p_\theta(z_t \mid z_{<t})$, where $z_{<t}$ represents the compressed L-OGM representations over the last t time steps, and θ are the network weights. The environment prediction task is inherently multimodal, and the latent vectors contributing to this stochasticity are unobservable. We introduce a variable s_t to capture this stochasticity at timestep t of the sequence and extend our model to $p_\theta(z_t \mid z_{<t}, s_{\leq t})$ [1]. During training, we extract the true posterior using an inference network $s_t \sim q_\phi(s_t \mid z_{\leq t})$ which has access to the z_t, representation of L-OGM at timestep t. While at test time, we sample from a learned prior $s_t \sim p_\gamma(s_t \mid z_{<t})$ as we are attemping to predict z_t. This process is autoregressively repeated for T_F future steps assuming access for T_O past observations. The framework is optimized using the variational lower bound objective [25]:

$$\begin{aligned}\mathcal{L} = -\sum_{t=T_O}^{T_O+T_F} & \mathbb{E}_{q_\phi\left(s_{\leq t} | z_{\leq t}\right)} \left[\log p_\theta\left(z_t \mid z_{<t}, s_{\leq t}\right)\right] \\ & + D_{KL}\left(q_\phi\left(s_t \mid z_{\leq t}\right) || \, p_\gamma\left(s_t \mid z_{<t}\right)\right),\end{aligned} \tag{3}$$

where the prediction network and prior autoregressively receive previous predicted embeddings, whereas the posterior takes in only the ground truth.

LOPR is implemented using a transformer decoder-based architecture, comprising a deterministic module $\mathcal{P}_\theta$ and two

inference networks $\mathcal{Q}_\phi$ and $\mathcal{Q}_\gamma$ for the prior and posterior, respectively. The positional information is provided through a sinusoidal positional encoding [46]. At each time step t, s_t is sampled from the inference network $\mathcal{Q}$, which outputs the parameters of the Gaussian distribution:

$$\mu,\ \sigma = \mathcal{Q}(z_{\leq \text{ or } <t}) \tag{4}$$

$$s_t \sim \mathcal{N}\left(\mu, \sigma^2\right), \tag{5}$$

where s_t is drawn from $\mathcal{Q}_\gamma$ at test time and from $\mathcal{Q}_\phi$ at training time, as explained above. Then z_{t-1} is attended with s_t and provided to the deterministic decoder where all past representations are incorporated:

$$z_t = \mathcal{P}_\theta(\text{SelfAttn}(z_{<t}, s_{\leq t})). \tag{6}$$

In the above operations, each $z_t \in \mathbb{R}^{c \times h \times w}$ is split along its spatial dimensions into k patches, which are then flattened [9]. Each token has dimension $\frac{chw}{k}$, thereby also facilitating spatial attention and optimizing the parameter count in the attention layer. This operation is applied to both the deterministic decoder and the inference networks. In the final step, the predicted compressed representations are concatenated, reshaped back to their original dimensions, and then provided to the decoder from Section III-A.

C. Diffusion-based Batch Decoder

We can decode each z_t independently using the single-frame decoder outlined in Section III-A to obtain high-quality predictions in real time. However, this approach can lead to poor temporal consistency and compression losses [19]. They manifest as unrealistic changes in the distribution of occupied cells over time and poor pixel-wise accuracy, particularly in the first predicted frames that should retain most of the static details from the observations.

We address these issues by refining $\hat{x}_{t-\Delta:t}$ from the single-frame decoder in batches with a diffusion-based batch decoder, where Δ is the number of frames. Δ is fixed and is not a function of batch size used at inference. At train time, the batch decoder is conditioned on ground truth decoded frames $\hat{x}_{t-\Delta:t}$ and a original rasterized frame preceding the sequence $x_{t-\Delta-1}$. We follow a standard video diffusion formulation that uses a 3D-UNet as a denoising model and minimizes the mean-squared error between the predicted and ground truth noises [18]. The model is trained to refine the decoded ground truth frames $\hat{x}_{t-\Delta:t}$ to more closely match $x_{t-\Delta:t}$. At test time, decoded frames and the preceding rasterized frame are reconstructed from predicted embeddings, with the exception of the first prediction, where the preceding frame is a rasterized observation, allowing static details to be preserved throughout the predicted sequence.

D. Conditioning on Other Sensor Modalities

LOPR can be conditioned on maps, the planned trajectory, and observed RGB camera images. We assume access to maps for the entire planned trajectory. Each input modality is first embedded as described below and then integrated into the framework using the self-attention mechanism before being provided to deterministic and inference networks.

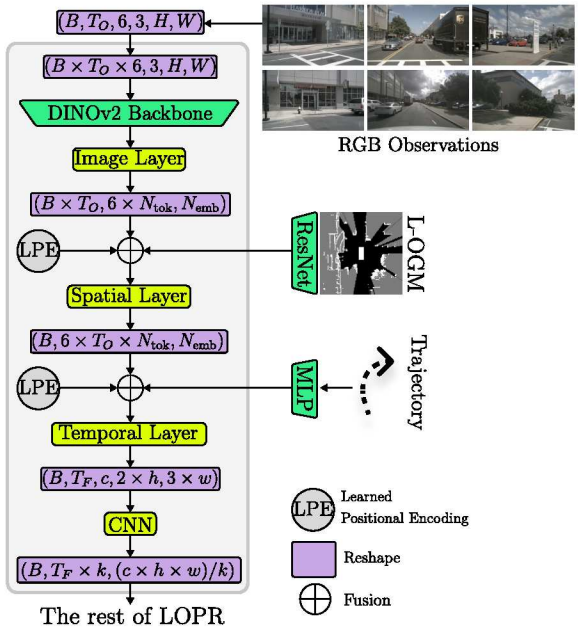

Fig. 3: Vision Transformer-based RGB Camera Encoder. RGB camera data is processed through the pre-trained DINOv2 backbone, subsequently passing through a series of attention layers. These layers aggregate information within each view (image layer), across different views (spatial layer), and throughout all observed timesteps (temporal layer). The spatial and temporal layers also include the learned positional embedding and are conditioned on the L-OGM embeddings and the planned trajectory, respectively.

Maps and Planned AV Trajectory. The map $m \in \mathbb{R}^{3 \times W \times H}$ comprises the drivable area, stop lines, and pedestrian crossings within the ego frame, represented using a rasterized format. The planned trajectory $tr \in \mathbb{R}^{3 \times T}$ includes position (x, y, z) for the entire sequence, normalized with respect to the ego position. Maps and the planned trajectory are processed with commonly used convolutional and fully connected networks.

RGB Camera. The camera observation, denoted as $c_t \in \mathbb{R}^{N \times 3 \times W \times H}$, encompasses views from N RGB cameras surrounding the vehicle. They offer important semantic information not available in L-OGMs. They can: 1) distinguish whether occupied cells are dynamic agents (like cars), including their type and orientation, or static environmental elements, and 2) provide insights into the state of the environment beyond the area observed in the fixed-size L-OGM. Incorporating RGB inputs into the self-supervised perception task is challenging due to the limited size of the AV perception datasets. To address this, we use the pretrained DINOv2 [34] backbone and finetune the module on a short deterministic L-OGM prediction task. The model is conditioned on observed L-OGMs and planned trajectory embeddings, which accelerates convergence. We also observe that adding trajectory embeddings significantly reduces uncertainty in future ego motion, enhancing the effectiveness of deterministic fine-tuning. This approach encourages the extraction of visual

information useful for downstream stochastic prediction. We note that DINOv2 is pre-trained on unlabeled data, aligning with the motivation of this work, unlike the commonly used ResNet-based backbones from object detection, which rely on manually labeled datasets.

Fig. 3 shows the image processing module $\mathcal{I}_\beta$. Each image is embedded using a DINOv2 backbone. It is followed by a series of attention modules: 1) The image layer, which aggregates tokens from a single view. 2) The spatial layer, which collects embeddings from each perspective around the vehicle along with a corresponding L-OGM embedding. 3) The temporal layer, which aggregates information across all observed time steps and the planned AV trajectory. We found that adding planned AV trajectory is beneficial but not necessary. Finally, the tokens from each time step are concatenated along the spatial dimensions and then processed through a convolutional layer to produce z_{cam}. The output tokens are then segmented back into patches, flattened, and integrated into the framework using the self-attention mechanism.

E. Conditioning on Other Information

Locations. Considering the diversity of locations in nuScenes [3], we append a one-hot location encoding (Singapore or USA) to the start of the observations sequence.

Sequence Augmentation. The open-source perception datasets are relatively small compared to vectorized trajectory datasets. To increase the number of samples, we implement a series of augmentations (e.g. mirror reflections, rotations, and reversing the sequence). Recognizing that these augmentations might adversely affect the prediction correctness (e.g. potentially resulting in predictions that do not adhere to driving rules), we attach a one-hot encoding of the augmentation type at the beginning of the sequence.

F. Prediction Extrapolation

We extend the prediction horizon at test time using a sliding-window approach, treating the last predicted frames as observations and repeating the process. However, unlike maps and planned trajectory modalities, we neither have access to nor predict future camera observations. Hence, we randomly drop out the image embeddings enabling robust extrapolation beyond the training-time prediction horizon.

IV. Experiments

We evaluate our framework by analyzing the pre-trained latent space, evaluating its performance in environment prediction tasks, and examining the impact of additional sensor modalities on the predictions quality.

A. Dataset

We use the nuScenes Dataset [3] and the Waymo Open Dataset [40]. nuScenes contains 5.5 hours of data collected in Boston and Singapore. Waymo Open Dataset [40] provides 6.4 hours of data compiled in San Francisco, Phoenix, and Mountain View. Both datasets include measurements from RGB cameras around the vehicle, LiDAR(s), and maps.

Data Representation: We generate L-OGMs in the ego vehicle frame using a ground-segmented LiDAR point cloud. The OGM dimensions are $H \times W = 128 \times 128$ with a $0.\overline{3}\,\text{m}$ resolution, corresponding to a $42.7\,\text{m} \times 42.7\,\text{m}$ grid. RGB images and maps are downscaled to 224×224 and 128×128 respectively. During sequence prediction training, we provide 5 past L-OGMs ($0.5\,\text{s}$) as observations alongside other sensor modalities, and forecast for 15 future frames ($1.5\,\text{s}$) at $10\,\text{Hz}$. We also extend the prediction horizon to 30 frames ($3.0\,\text{s}$) to evaluate the extrapolation capabilities of our framework.

B. Architecture and Training Details

Architectures. We incorporate a convolutional network for all modules except the transformer-based ones and the trajectory encoder. The discriminator architecture is multi-scale and multi-patch, inspired by prior work [20]. The dimension of the latent vector is set at $z \in \mathbb{R}^{64\times4\times4}$ which is split in 4 patches resulting in the flattened embedding size of 256. For the prediction network, the decoder and variational modules each consist of 6 layers and 6 heads, collectively comprising 16.1 million parameters. The image module employs a DINOv2 ViT-S/14 backbone [34]. Within this module, the image, spatial, and temporal layers comprise 1, 4, and 4 layers, respectively, each with 4 heads. The total number of parameters for the image head is 27.4M. For the diffusion-based decoder, we leverage an implementation by HuggingFace [47]. In the results section, we use a single-step decoder, unless stated otherwise.

Model Training. We used the AdamW optimizer [29] with $lr = 4 \times 10^{-4}$. For representation learning, we used three NVIDIA TITAN X 24 GB for 360k steps with a total batch size of 30. We trained the L-OGM-only prediction models on a single NVIDIA TITAN RTX 24GB GPU and the multimodal model and batch decoder on two NVIDIA L40 48GB GPUs. The models were trained with a total batch size of 40 until convergence. For the stochastic prediction component, we use a KL annealing with $\beta = 2 \times 10^{-6}$ for the first 10 epochs followed by a linear increase to 0.2 over 50k training steps.

C. Evaluation

Baselines. We benchmark our approach against methods commonly used in L-OGM prediction, including PredNet [30, 21] and TAAConvLSTM [26]. Additionally, we compare against OccWorld [52], a state-of-the-art network for 3D semantic occupancy prediction adapted for the 2D L-OGM task. OccWorld uses the latent space of VQ-VAE and incorporates a customized transformer to enhance inference efficiency. To ensure a fair comparison, we increase its latent space size to match the reconstruction performance of our method. We further evaluate our approach against state-of-the-art real-time video prediction models, including SimVP V2 [13], PredRNN V2 [49], and E3DLSTM [48]. We train all models until convergence. As an additional baseline, we include a naive method that repeats the last observed frame across the entire prediction horizon, providing a reference for assessing the models' ability to capture the scene's motion dynamics.

Metric: We evaluate all models using the Image Similarity (IS) metric [2] across the 1.5 s and 3.0 s prediction horizons. For stochastic predictions, we sample 10 predictions and evaluate the best one. IS calculates the smallest Manhattan distance between two grid cells with the same thresholded occupancy, capturing the spatial error of predictions. It determines the grid distance between matrices m_1 and m_2 [2]:

$$\psi(m_1, m_2) = \sum_{c \in \mathcal{C}} d(m_1, m_2, c) + d(m_2, m_1, c) \tag{7}$$

where

$$d(m_1, m_2, c) = \frac{\sum_{m_1[p]=c} \min\{\mathrm{md}(p_1, p_2) | m_2[p_2] = c\}}{\#_c(m_1)}. \tag{8}$$

The set of discrete values $\mathcal{C}$ possibly assumed by m_1 or m_2 are: occupied, occluded, and free. $m_1[p]$ denotes the value c of map m_1 at position $p = (x, y)$. $\mathrm{md}(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2|$. $\#_c(m_1) = \#\{p_1 \mid m_1[p_1] = c\}$ is the number of cells in m_1 with value c.

V. Results

A. Latent Space Analysis

The latent space trained during the representation learning stage is crucial for facilitating accurate predictions. If an agent in the observed frames is lost during the encoding phase, the prediction network struggles to recover this information, leading to incorrect forecasts. We examine the impact of different representation learning losses on reconstruction and prediction performance in Table I. Our results highlight a trade-off between single-grid reconstruction performance and prediction performance. The simple autoencoder performs well in reconstruction and short-term prediction but suffers in prediction accuracy as the horizon extends. In contrast, incorporating KL and adversarial components reduces reconstruction performance but enhances long-term prediction capability, aligning with the primary objective of this work. We also report the performance of our representation learning approach with all augmentations, demonstrating its positive impact on all metrics. For context, we provide randomly selected L-OGMs and their reconstructions in Fig. 4, which illustrate that the encoder-decoder setup effectively reconstructs the scenes.

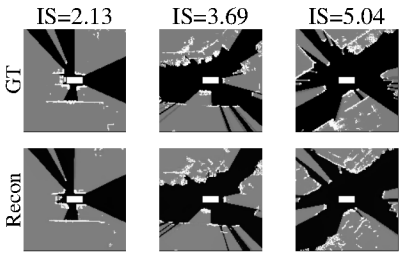

Fig. 4: Our encoder-decoder effectively reconstructs the L-OGMs, with the IS score reflecting differences in the distribution of occupied cells (e.g., IS of 2.13 and 3.69). In rare instances (e.g., rotation with many agents, IS=5.04), it might lose some detail. White represents occupied cells, black denotes free space, and shades of gray capture values between 0 and 1.

TABLE I: The impact of representation learning objective on the reconstruction ($\mathbf{IS}_{recon}$) and prediction performance ($\mathbf{IS}_{5\to X}$) on nuScenes with no augmentation during representation learning. VAE-GAN (KL + Adv) excels at long-term prediction horizon. We also report VAE-GAN performance with the augmented dataset.

Recon	KL	Adv	$\mathbf{IS}_{recon}(\downarrow)$	$\mathbf{IS}_{5\to15}(\downarrow)$	$\mathbf{IS}_{5\to30}(\downarrow)$
No augmentations in representation learning					
✓	×	×	**2.18 ± 0.01**	**7.88 ± 0.16**	12.18 ± 0.29
✓	✓	×	5.90 ± 0.01	9.76 ± 0.15	12.69 ± 0.20
✓	×	✓	4.36 ± 0.01	8.36 ± 0.14	11.72 ± 0.21
✓	✓	✓	4.82 ± 0.01	7.94 ± 0.12	**10.56 ± 0.17**
With augmentations in representation learning					
✓	✓	✓	3.30 ± 0.01	7.00 ± 0.10	9.76 ± 0.16

TABLE II: Quantitative evaluation of the prediction performance. The best results are bolded, and the second-best results are underlined. † indicates a stochastic model. In *Ours*, + indicates the addition of a feature to a model that includes all modifications listed in the rows above. We also report the frequency and CUDA memory consumption for generating a single 15-frame prediction.

Model	NuScenes Dataset		Waymo Open Dataset			
	$\mathbf{IS}_{5\to15}$	$\mathbf{IS}_{5\to30}$	$\mathbf{IS}_{5\to15}$	$\mathbf{IS}_{5\to30}$	T (Hz)	M (GB)
PredRNN V2	30.69±2.10	79.95±4.07	28.45±1.98	62.71±3.38	7.69	0.23
Sim.VP V2	20.02±1.28	47.20±2.63	15.87±1.18	46.38±2.66	**37.04**	0.26
ED3LSTM	10.33±0.29	19.49±0.71	14.36±1.04	36.60±2.15	5.55	0.94
TAAConvLSTM	7.02±0.17	15.26±0.66	6.43±0.36	20.51±1.39	5.65	0.11
PredNet	7.10±0.19	13.93±0.44	6.78±0.41	22.38±1.54	16.39	**0.10**
OccWorld-2D†	7.84±0.09	11.90±0.18	6.72±0.13	11.03±0.26	3.00	1.10
Fixed Frame	11.50±0.14	14.41±0.18	10.35±0.41	14.74±0.54	-	-
Ours						
Deterministic	7.94±0.13	11.48±0.22	7.62±0.33	12.12±0.73	<u>25.64</u>	2.49
+Aug.	7.24±0.11	10.47±0.19	7.27±0.37	11.81±0.74	<u>25.64</u>	2.49
+Stoch.†	7.00±0.10	9.76±0.16	6.64±0.19	9.93±0.28	11.91	2.51
+R+T+M†	<u>6.36±0.08</u>	<u>8.32±0.12</u>	<u>6.23±0.18</u>	<u>9.00±0.28</u>	9.09	2.61
+Diff. Dec.†	**4.88±0.09**	**7.12±0.12**	**5.24±0.18**	**8.17±0.27**	<0.01	17.99

B. Prediction Task

General Performance: We compare our framework against the baselines in Table II. The integration of dataset augmentations, stochasticity, additional input modalities, and a diffusion decoder each contribute to notable improvements. LOPR outperforms previous methods on both datasets, with improvements becoming more pronounced as the prediction horizon extends. We visualize examples of predictions rolled out for 3.0 s, extending beyond the prediction horizon used during training, in Figs. 5 and 6. Our framework generates high-quality, realistic predictions, supporting the quantitative results while remaining real-time feasible.

Stochasticity: Stochasticity in the L-OGM primarily stems from the unknown intents of other agents in the scene and partial observations. Our framework models the multimodal distribution of future agents' positions. It is capable of inferring hypothetical, previously unobserved agents (see Fig. 7) and capturing the varying motion dynamics of observed agents (see Figs. 5 to 7). As shown in Table II, modeling stochasticity has a positive quantitative impact compared to the deterministic framework. The variational module enables high-quality and semantically meaningful multi-future reasoning about the behaviors of both observed and unobserved agents.

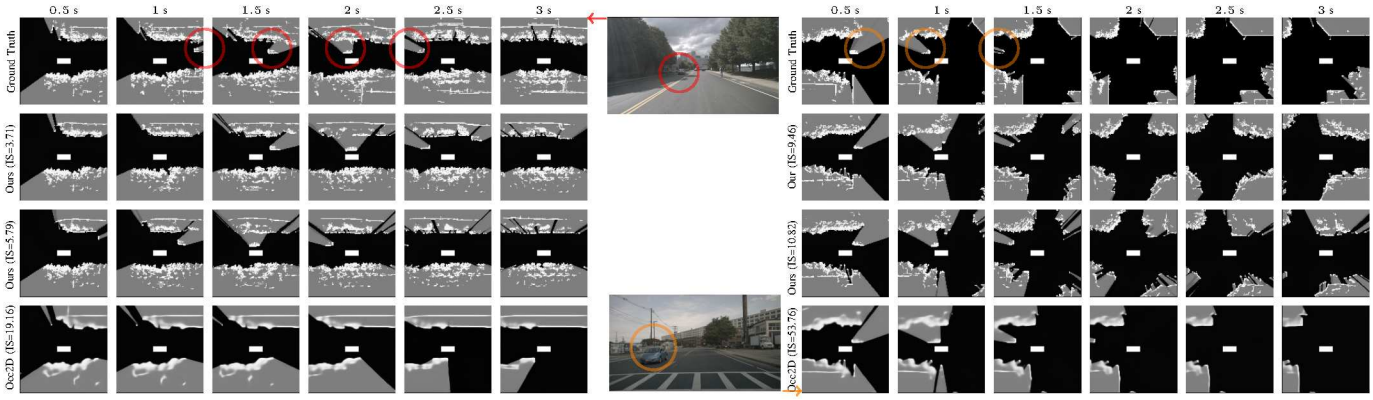

Fig. 5: Examples of LOPR and OccWorld-2D predictions with visualized front camera observations from the nuScenes dataset. The predictions are generated with the single-step decoder. LOPR is conditioned on all cameras around the vehicle, maps, and the planned trajectory. We report IS scores for each sample. (Left) Prediction of an oncoming vehicle (red) visible only in the front camera. Each LOPR sample captures a realistic hypothetical evolution of the scene, such as variations in the velocity of the oncoming car. (Right) Correct forecasting of an oncoming vehicle and a static road layout (orange). Both examples demonstrate that our framework is capable of multi-future reasoning and leveraging multi-modal observations.

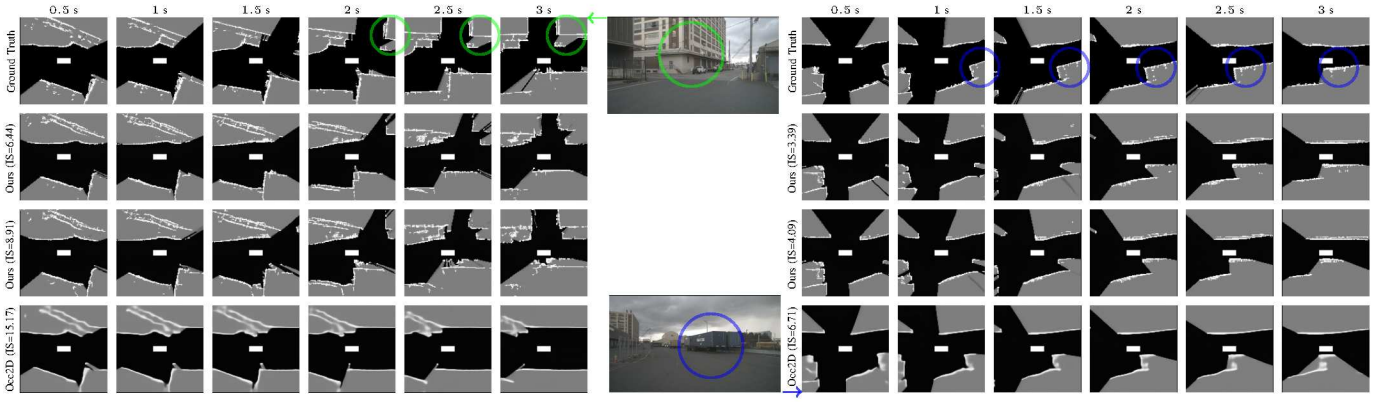

Fig. 6: Examples of LOPR and OccWorld-2D predictions with visualized front camera from the nuScenes dataset. The prediction setting is the same as in Fig. 5. (Left) Accurate prediction of the static road layout and a parked vehicle (green) visible only in the front camera. Each LOPR sample provides a realistic occupancy representation of the parked car and the environment's layout. (Right) Accurate prediction of a parked truck (blue).

Impact of trajectory, map, and camera conditioning: LOPR leverages additional input modalities to enhance its understanding of the surroundings and the ego vehicle's intent, enabling more accurate predictions. This includes accurately inferring static elements of the environment, such as road layouts and parked vehicles, as well as dynamic components like oncoming vehicles. These features, while visible in the cameras and maps, lie beyond the observable area in L-OGMs (see Figs. 5 and 6). In Table III, we evaluate the impact of incorporating trajectory conditioning, maps, and cameras as input modalities. Our results demonstrate that each modality contributes to significant numerical improvements, with the best performance achieved when all available input modalities are combined. Furthermore, we observe a positive impact of DINO-based camera module finetuning on the deterministic occupancy prediction task.

Diffusion-based Decoder: Known weaknesses of making predictions in the latent space of generative models include poor temporal consistency and compression losses. We address these concerns with a diffusion-based batch decoder which leads to significant improvements in temporal consistency and the maintenance of detail that is often lost due to compression, as shown in Fig. 8. Numerically, it also results in significant improvements, especially in short-horizon predictions, due to

TABLE III: Impact of trajectory, map, and camera modalities on the prediction performance. $*$ denotes a camera module without finetuning on the deterministic task. All models were trained for 40 epochs.

Traj.	Maps	Cameras	$\mathbf{IS}_{5\rightarrow 15}(\downarrow)$	$\mathbf{IS}_{5\rightarrow 30}(\downarrow)$
×	×	×	7.20 ± 0.11	10.02 ± 0.16
✓	×	×	6.75 ± 0.11	8.94 ± 0.14
×	✓	×	6.88 ± 0.10	9.18 ± 0.14
×	×	✓	6.91 ± 0.10	9.54 ± 0.15
×	×	✓*	7.20 ± 0.11	9.98 ± 0.16
✓	✓	×	6.73 ± 0.10	8.86 ± 0.14
✓	×	✓	6.47 ± 0.09	8.61 ± 0.13
×	✓	✓	6.62 ± 0.09	8.83 ± 0.13
✓	✓	✓	**6.44 ± 0.09**	**8.50 ± 0.12**

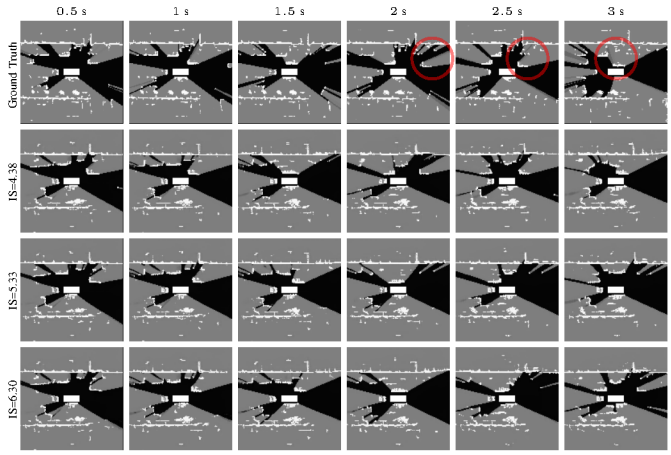

Fig. 7: Visualization of stochastic predictions using a single-step decoder conditioned solely on L-OGMs, with the IS score averaged over the entire sequence. The scenario depicts multiple agents moving along a road, with the bottom three rows (alongside their IS scores) representing different samples from the variational model. Our model captures diverse motion patterns of observed vehicles and can even infer the presence of unobserved ones. It is marked with red in the ground truth and correctly inferred in the first sample (IS = 4.38).

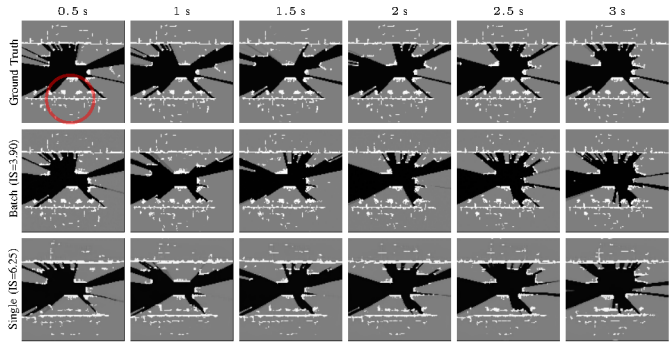

Fig. 8: Visualization of generated predictions decoded with a single-step decoder (IS=6.25) and further refined with a diffusion-based batch decoder (IS=3.90). The batch decoder increases temporal consistency between frames and reduces compression losses. In red, we highlight trees on the side of the road that are recovered with our batch decoder.

the preservation of static details. However, this comes at the cost of real-time feasibility. While our approach with the single-step decoder and all baselines generate a full sequence of predictions at a rate of $3-38$ Hz, decoding predictions with a diffusion decoder takes minutes. Although it is not currently feasible for real-time applications, recent advancements in diffusion-based consistency models [39] suggest a promising path toward achieving this.

Real-time Feasibility: In Table II, we provide the frequency of generating a single 15-frame prediction in full precision and the CUDA memory consumption for context. All models are tested on NVIDIA L40. Our results demonstrate that LOPR with a single-step decoder is real-time feasible, achieving 11.91 Hz without additional input modalities and 9.09 Hz when all input modalities are used. In comparison, OccWorld-2D, which operates in the latent space of VQ-VAE and uses a customized transformer to accelerate inference, achieves 3.00 Hz. For further context, vanilla transformer applied to a similar size discrete latent space and observation-prediction horizon requires 30 seconds to generate a single prediction on an NVIDIA V100 [36].

Baseline Comparison: There are several reasons for the significant improvements over prior work. Most prior work is optimized for deterministic prediction tasks in grid cell space and fails to condition on the input modalities available in the autonomous vehicle stack. These methods do not capture the stochastic nature of motion forecasting and lack semantic information about the scene (e.g., distinguishing whether an occupied cell is due to a pedestrian or a sign). As a result, their forecasted L-OGMs gradually lose important details over the prediction horizon, leading to the disappearance of moving objects. While they might capture some static details of the scene, resulting in respectable IS scores for short prediction horizons, they ultimately yield high IS scores as static details vanish and deterministic predictions prove insufficient. Our numerical results demonstrate that the deterministic variation of our framework, conditioned on L-OGMs (the same setting as the baselines), outperforms all deterministic baselines over extended prediction horizons. As we incorporate multi-future reasoning, provide additional semantic conditioning, and introduce diffusion decoding, the performance gap between LOPR and the baselines becomes even more substantial, as shown in Table II. Interestingly, our work also outperforms the state-of-the-art in semantic occupancy prediction, which uses a customized transformer to predict discrete latent representations. This highlights an alternative to the commonly used transformer with discrete codebooks, which often incurs significant cost at inference time.

VI. Limitations

LOPR relies on a well-trained latent space, which may sometimes lose critical details necessary for accurate predictions. If the latent encoding misses important information, the prediction network may not recover it, leading to potentially inaccurate predictions. We have partially addressed this issue with a diffusion decoder. However, this approach may still result in inaccuracies and is not real-time feasible. Additionally, the performance of our framework is heavily influenced by the size of the available perception datasets. The datasets used in this paper contain about six hours of data, which limits performance in scenarios not well-represented in the dataset, such as intersection interactions and cross-traffic.

VII. Conclusion

In this paper, we proposed a self-supervised L-OGM prediction framework that captures the stochasticity of the scene and is conditioned on multi-modal observations available in autonomous vehicles. LOPR consists of a VAE-GAN-based generative model that learns an expressive low-dimensional latent space, and a transformer-based stochastic prediction network that operates on this continuous latent space. Our experiments demonstrate that LOPR outperforms all prior

approaches both qualitatively and quantitatively while maintaining real-time feasibility. Furthermore, it surpasses commonly used transformer-based methods with discrete representations, while offering significantly faster inference. We also highlight the advantages of incorporating trajectory, map, and camera conditioning, which enhance the framework's capabilities. Additionally, we extend the framework with a diffusion decoder to address temporal consistency issues and reduce compression losses, albeit with a trade-off in real-time feasibility. In future work, we will explore extending LOPR to perform 3D occupancy prediction, and apply it to other tasks, such as occlusion inference [22, 27] and path planning.

ACKNOWLEDGMENTS

This project was made possible by funding from the Ford-Stanford Alliance.

REFERENCES

[1] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *International Conference on Learning Representations (ICLR)*, 2018.

[2] Andreas Birk and Stefano Carpin. Merging occupancy grid maps from multiple robots. *Proceedings of the IEEE*, 94(7):1384–1397, 2006.

[3] 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 *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631. IEEE, 2020.

[4] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Conference on Robot Learning (CoRL)*, 2020.

[5] Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Scept: Scene-consistent, policy-based trajectory predictions for planning. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17103–17112, 2022.

[6] Chiho Choi, Joon Hee Choi, Jiachen Li, and Srikanth Malla. Shared cross-modal trajectory prediction for autonomous driving. In *cvpr*, pages 244–253, 2021.

[7] Harrison Delecki, Masha Itkina, Bernard Lange, Ransalu Senanayake, and Mykel J Kochenderfer. How do we fail? stress testing perception in autonomous vehicles. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 5139–5146. IEEE, 2022.

[8] Julie Dequaire, Peter Ondrúška, Dushyant Rao, Dominic Wang, and Ingmar Posner. Deep tracking in the wild: End-to-end tracking using recurrent neural networks. *International Journal of Robotics Research*, 37(4-5):492–512, 2018.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[10] Mariella Dreissig, Dominik Scheuble, Florian Piewak, and Joschka Boedecker. Survey on lidar perception in adverse weather conditions. In *Intelligent Vehicles Symposium (IV)*, 2023.

[11] Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[12] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *International Conference on Computer Vision (ICCV)*, pages 9710–9719, 2021.

[13] Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. Simvp: Simpler yet better video prediction. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3170–3180, 2022.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems (NeurIPS)*, 27, 2014.

[15] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[16] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2019.

[17] Geoffrey E Hinton and Richard Zemel. Autoencoders, minimum description length and Helmholtz free energy. *Advances in Neural Information Processing Systems (NeurIPS)*, 6, 1993.

[18] Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022.

[19] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

[20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017.

[21] Masha Itkina, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Dynamic environment prediction in urban scenes using recurrent representation learning. In *International Conference on Intelligent Transportation Systems (ITSC)*, pages 2052–2059. IEEE, 2019.

[22] Masha Itkina, Ye-Ji Mun, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. Multi-agent variational occlusion inference using people as sensors. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

[23] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8110–8119, 2020.

[24] Seung Wook Kim, Jonah Philion, Antonio Torralba, and Sanja Fidler. DriveGAN: Towards a controllable high-quality neural simulation. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5820–5829, 2021.

[25] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations (ICLR)*, 2014.

[26] Bernard Lange, Masha Itkina, and Mykel J Kochenderfer. Attention augmented ConvLSTM for environment prediction. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1346–1353. IEEE, 2020.

[27] Bernard Lange, Jiachen Li, and Mykel J Kochenderfer. Scene informer: Anchor-based occlusion inference and trajectory prediction in partially observable environments. In *International Conference on Robotics and Automation (ICRA)*, 2024.

[28] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning (ICML)*, pages 1558–1566, 2016.

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[30] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *International Conference on Learning Representations (ICLR)*, 2017.

[31] Reza Mahjourian, Jinkyu Kim, Yuning Chai, Mingxing Tan, Ben Sapp, and Dragomir Anguelov. Occupancy flow fields for motion forecasting in autonomous driving. *IEEE Robotics and Automation Letters*, 7(2):5639–5646, 2022.

[32] Nima Mohajerin and Mohsen Rohani. Multi-step prediction of occupancy grid maps with recurrent neural networks. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10600–10608. IEEE, 2019.

[33] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S. Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *International Conference on Robotics and Automation (ICRA)*, pages 2980–2987, 2023.

[34] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. 2024.

[35] Charles R Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3d object detection from point cloud sequences. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6134–6144, 2021.

[36] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. In *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2021.

[37] Marcel Schreiber, Stefan Hoermann, and Klaus Dietmayer. Long-term occupancy grid prediction using recurrent neural networks. In *International Conference on Robotics and Automation (ICRA)*, pages 9299–9305. IEEE, 2019.

[38] Marcel Schreiber, Vasileios Belagiannis, Claudius Gläser, and Klaus Dietmayer. Dynamic occupancy grid mapping with recurrent neural networks. In *International Conference on Robotics and Automation (ICRA)*, pages 6717–6724. IEEE, 2021.

[39] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[40] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo Open Dataset. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454. IEEE, 2020.

[41] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[42] Maneekwan Toyungyernsub, Masha Itkina, Ransalu Senanayake, and Mykel J Kochenderfer. Double-prong ConvLSTM for spatiotemporal occupancy prediction in dynamic environments. In *International Conference on Robotics and Automation (ICRA)*, pages 13931–13937. IEEE, 2021.

[43] Maneekwan Toyungyernsub, Esen Yel, Jiachen Li, and Mykel J Kochenderfer. Dynamics-aware spatiotemporal occupancy prediction in urban environments. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 10836–10841. IEEE, 2022.

[44] Maneekwan Toyungyernsub, Esen Yel, Jiachen Li, and Mykel J Kochenderfer. Predicting future spatiotemporal

occupancy grids with semantics for autonomous driving. In *Intelligent Vehicles Symposium (IV)*, 2024.

[45] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

[47] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

[48] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *International Conference on Learning Representations (ICLR)*, 2019.

[49] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, S Yu Philip, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2208–2225, 2022.

[50] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

[51] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

[52] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.