Neural Volumetric World Models for Autonomous Driving

Zanming Huang*, Jimuyang Zhang*, and Eshed Ohn-Bar

Boston University
{huangtom,zhangjim,eohnbar}@bu.edu

Abstract. Effectively navigating a dynamic 3D world requires a comprehensive understanding of the 3D geometry and motion of surrounding objects and layouts. However, existing methods for perception and planning in autonomous driving primarily rely on a 2D spatial representation, based on a bird's eye perspective of the scene, which is insufficient for modeling motion characteristics and decision-making in real-world 3D settings with occlusion, partial observability, subtle motions, and varying terrains. Motivated by this key insight, we present a novel framework for learning end-to-end autonomous driving based on volumetric representations. Our proposed neural volumetric world modeling approach, NeMo, can be trained in a self-supervised manner for image reconstruction and occupancy prediction tasks, benefiting scalable training and deployment paradigms such as imitation learning. Specifically, we demonstrate how the higher-fidelity modeling of 3D volumetric representations benefits vision-based motion planning. We further propose a motion flow module to model complex dynamic scenes, enabling additional robust spatiotemporal consistency supervision. Moreover, a temporal attention module is introduced to effectively integrate predicted future volumetric features for the planning task. Our proposed sensorimotor agent achieves state-of-the-art driving performance on nuScenes and CARLA, outperforming prior baseline methods by over 18%.

Keywords: Autonomous Driving · Self-Supervised Pre-training · Planning

1 Introduction

Navigation in the 3D world requires a comprehensive understanding of dynamic 3D surroundings [4,29,33]. Detailed modeling of the shape and motion of 3D objects is particularly crucial when navigating in intricate, safety-critical settings such as autonomous driving. For instance, careful reasoning over a myriad of surrounding 3D characteristics, including frequent partial occlusions, uneven surfaces, off-ground objects and their shapes, and subtle motion and maneuvers by vehicles signaling future intent, may all mean the difference between seamless and safe navigation or a wrong maneuver with potentially dire consequences.

Despite recent advancements in modeling for specific 3D tasks for autonomous driving, such as 3D segmentation and scene flow [11, 14, 19, 40, 54, 56, 60, 72, 80, 82], how these tasks and models may be tightly integrated to facilitate the final driving

* Equally contributed.

task [8, 24, 43, 67] remains an open question. For instance, most sensorimotor (vision-to-decision) models rely on simplified and inadequate representations of the 3D world, i.e., in the Bird's Eye View (BEV), to couple perception with action while integrating motion prediction and planning tasks [17, 23, 44, 46, 75, 85]. However, the planar BEV lacks expressiveness, as it can only provide a coarse representation of the volumetric and dynamic 3D world. Additionally, current methods hinder scalability, as they leverage extensive supervision (either as part of a modular pipeline or as an auxiliary task, with known semantic segmentation targets in the BEV). In contrast, humans and animals can adeptly navigate complex and dynamic 3D environments by leveraging *self-supervised* spatial and predictive representations [29, 30, 34, 39, 45, 53, 70, 73].

Given the naive modeling of 3D geometry and temporal aspects by existing visuomotor models, we aim to develop more generalized and scalable frameworks for efficiently encapsulating 3D structures and their dynamics. We focus on a scalable, self-supervised architecture and training process that does not rely on extensive and cumbersome manual 3D annotations while jointly optimizing for the ultimate driving decision-making task. Our key insight lies in leveraging recent advancements in 3D world modeling, particularly based on neural rendering [31, 49, 52, 62], noting that these are rarely explored as functional representations for making autonomous driving *decisions*. Moreover, the aforementioned frameworks generally study static scenes, whereas we emphasize dynamic and dense 3D settings. Nonetheless, they may still provide a useful geometric prior, i.e., as an auxiliary and consistency-based self-supervised reconstruction task.

Contributions: We introduce NeMo, a 3D volumetric based end-to-end sensorimotor driving framework enabled by three key components: (1) Self-supervised volumetric representation pre-training with image reconstruction and occupancy prediction tasks through neural rendering. (2) A motion flow module for modeling dynamic scenes in complex urban driving, leveraging spatiotemporal consistency for additional supervision. (3) A volumetric planner with a temporal attention module that effectively fuses predicted future features for motion planning. In our experiments, we achieve state-of-the-art performance in open-loop evaluation in the nuScenes benchmark, improving over prior baseline methods by over 18% in performance.

2 Related Work

2.1 Learning-based Motion Planning

Learning-based end-to-end driving systems are garnering increasing attention in the research due to their simplicity and impressive performance. Existing approaches can be classified into two categories: imitation learning (IL) based methods and reinforcement learning (RL) based methods. In IL [5, 6, 8, 25, 86, 87], an agent is trained by imitating the behavior of an expert. RL, on the other hand, can train the agent using a reward signal from trial and error [15, 16, 48, 64, 74, 79, 89]. To enhance the interpretability of such end-to-end systems, recent approaches introduce intermediate learning tasks [7]. For instance, several methods learn features in a BEV space [17, 85], i.e., based on introduced BEV perception tasks. UniAD [43] effectively unifies perception, prediction,

and planning tasks through a query-based design, leading to impressive performance in planning. VAD [46] encodes image features into BEV space, which are used to learn the vectorized scene representation for motion planning. However, most methods leverage a simplified BEV space, which cannot capture intricate and complex characteristics. Although BEV features have shown remarkable simplicity and effectiveness, in our work we demonstrate that learning features in 3D volumetric space provides more fine-grained information, leading to improved planning performance.

2.2 Self-supervised Visual Representation Learning

Self-supervised learning has demonstrated great potential in real-world applications due to its ability to scale and adapt to new situations without human effort. Past works may bolster visual feature learning through motion and actions, such as moving through the scene or manipulating objects in view [2, 66]. The changes in view induced by these actions are then used as supervisory signals for training. Moreover, recent works further impose contrastive losses between manually augmented inputs [13, 20, 36]. However, the feature representations of the aforementioned methods are learned by enforcing constraints in the 2D image plane. In contrast, our proposed work extends the learned feature representations to 3D space given image inputs, which naturally contain richer information and can be used for downstream tasks that require extensive spatial reasoning, such as motion planning. Self-supervised 3D representations can be learned through motion, visual cues, or spatial consistency cues, e.g., [32, 41, 51, 57, 68]. Khurana et al. [49] learns geometric occupancy using LiDAR self-supervision. Gkioxari et al. [32] reconstruct 3D scenes by exploiting consistency between different viewing angles. However, their work considers simple indoor scenes, while our work addresses complex outdoor scenarios. Lai et al. [52] predict 3D features alongside the prediction of ego motions between frames. The features are transformed into future frames by ego-motion to reconstruct future image frames. Consistency loss and reconstruction loss are introduced to enforce the spatial coherence of the learned features. However, the work assumes a static environment, as dynamic objects would violate spatial consistency after transformation. This makes the method unsuitable for complex dynamic scenarios, such as driving on open roads. A recent study by ViDAR [83] analyzes learning 3D representations via a latent rendering operator for future point cloud prediction. In contrast to this concurrent work, NeMo introduces a motion flow module that better captures the 3D motion of objects in intricate dynamic scenarios. Moreover, we propose leveraging additional supervision from RGB images through neural volumetric rendering. Our experiments demonstrate the effectiveness of the proposed methods for 3D representation learning and downstream planning task.

2.3 Pre-training for Autonomous Driving

In our work, we focus on learning effective feature representation through self-supervision. Pre-training strategies have been widely used in computer vision. Features pre-trained on a comprehensive dataset like ImageNet [26] exhibit effective transferability across diverse settings and tasks. Moreover, studies have demonstrated the effectiveness of

pre-training with weakly-labeled datasets [58, 71, 81]. Our recent line of methods includes leveraging self-supervised contrastive learning objectives [21, 37]. Inspired by BERT [27], some works employ masked image reconstruction as a self-supervised pre-training approach [35, 76]. However, pre-training in the context of end-to-end autonomous driving remains under-discussed [50, 88]. PPGeo [78] pre-trains an effective visual encoder by predicting the ego-motion and minimizing the photometric error based on visual observations. ViDAR [83] pre-trains the model through a visual point cloud forecasting task for general autonomous driving. In contrast, our NeMo approach learns a more effective volumetric representation through image reconstruction using neural rendering and occupancy prediction. These newly introduced supervision signals, along with the novel motion flow module, are shown to benefit the ego motion planning task.

3 Method

Towards robust and scalable planning in a dynamic 3D world, the proposed NeMo approach effectively learns a volumetric representation through self-supervision on readily available driving data without extensive manual annotations. We further show such learned volumetric representation to be beneficial for downstream planning task. In this section, we first introduce an overview of NeMo in Sec. 3.1. Next, we introduce the process of learning the volumetric scene representation in Sec 3.2. Finally, we detail our approach to planning with the learned representation in Sec 3.3. An overview of our complete framework is shown in Fig. 1.

3.1 Overview

The overall framework of NeMo is shown in Fig. 1. Our framework follows a two-stage process. In the first stage, given multi-view RGB image inputs, NeMo first encodes them into volumetric features using a transformer-based feature encoder network. This encoder extracts and maps image features to their corresponding 3D space with camera parameters. The extracted features are supervised via image reconstruction and occupancy prediction across current and future frames using a neural volumetric rendering technique, which enables learning fine-grained feature representations and ensuring spatial and temporal consistency. In the second stage, planning is performed by leveraging the learned volumetric representation.

3.2 Volumetric Representation Learning

Volumetric Feature Encoding: To generate expressive spatial features for encoding 3D geometric and semantic information, we employ a transformer-based module that refines 3D spatial features by attending to specific regions in the RGB image input. Our 3D deformable attention network is derived from Zhu et al. [92]. Specifically, given n view image input $\mathbf{I}_t \in \mathbb{R}^{n \times H \times W \times 3}$ at timestep t, we obtain volumetric features

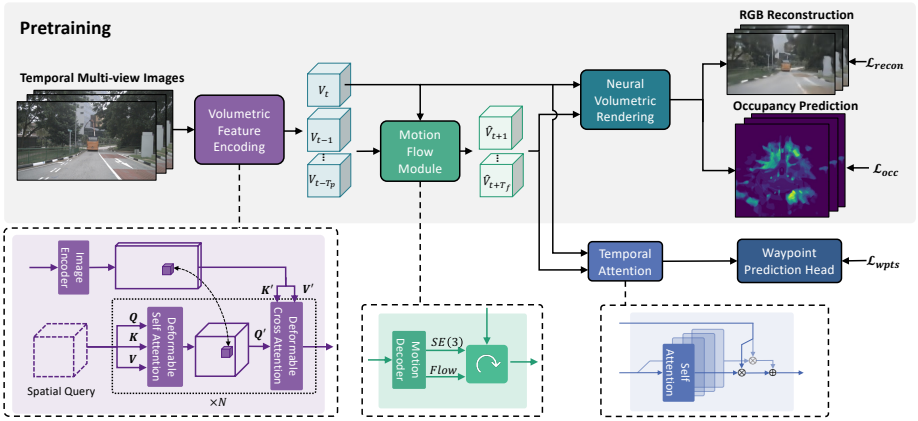

Fig. 1: Network Architecture and Training Process of NeMo. The temporal multi-view camera images are encoded into spatial volumetric features through a transformer-based attention module (colored in cyan blue). The motion flow module takes the temporal volumetric features, and predicts feature motion flow and ego motion that transform current volumetric features into future T_p steps (colored in green). Then, the neural volumetric rendering module renders RGB and occupancy for each volumetric feature (colored in purple). For motion planning, the temporal attention module fuses the predicted future volumetric features with the current ones (colored in blue), which are used for motion planning (colored in red). Our training is done in two stages. In the pre-training stage, the model is trained through self-supervision with RGB reconstruction and occupancy prediction task. In the fine-tuning stage, an additional waypoint prediction task is applied, i.e., learning a planner via behavioral cloning.

$\mathbf{V}_t \in \mathbb{R}^{X \times Y \times Z \times D}$ through multiple deformable self-attention and cross-attention layers. In particular, the cross-attention mechanism projects the 2D image features to 3D volumetric space and can be described as,

$$\mathbf{V}_t = \text{DeformAttn}(\mathbf{F}, \mathcal{T}(\mathbf{z}), \mathbf{I}_t) \tag{1}$$

where $\mathbf{F} \in \mathbb{R}^{X \times Y \times Z \times D}$ is a learned volumetric feature query, $\mathbf{z} \in \mathbb{R}^3$ is the coordinate of a location in the volumetric space and $\mathcal{T}(\mathbf{z})$ projects $\mathbf{z}$ to a corresponding coordinate in the input images using camera parameters. We use $\mathcal{T}(\mathbf{z})$ as reference points for the deformable attention mechanism, which aligns 2D image features with 3D volumetric features. We note that our attention module performs 2D-to-3D alignment, in contrast to 2D-to-2D attention-based alignment methods [19, 55, 90]. The resulting 3D features can be trained more effectively for the downstream auxiliary reconstruction tasks (i.e., RGB reconstruction and occupancy prediction, as shown in Fig. 1). Additional details are included in the supplementary.

Neural Volumetric Rendering: As shown in Fig. 2, we adopt a neural volumetric rendering approach for image reconstruction and occupancy prediction. To introduce our approach, we begin by parameterizing camera rays and expressing the points along the camera ray as

$$\mathbf{x} = \mathbf{o} + \lambda \mathbf{d} \tag{2}$$

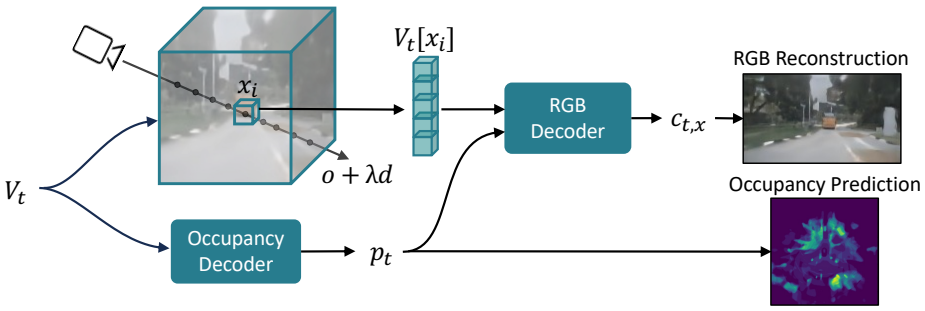

Fig. 2: Neural Volumetric Rendering. We show the neural volumetric rendering operation at timestep t for the ray $\mathbf{x}$. First, an occupancy decoder takes the volumetric feature $\mathcal{V}_t$ as input and predicts the occupancy $\mathbf{p}_t$. Subsequently, N points $\{\mathbf{x}_0, \ldots, \mathbf{x}_N\}$ are sampled along the camera ray x and are passed to the RGB decoder. The sampled features and predicted occupancy are used jointly to render the feature for ray $\mathbf{x}$ (Eqn. 6). Finally, the rendered features are then used to reconstruct the RGB image.

Where $\mathbf{o} \in \mathbb{R}^3$ is the ray origin, λ represents the distance along the ray, and $\mathbf{d} \in \mathbb{R}^3$ being the ray direction. For the sake of simplicity, we omit the index for each ray. Further, the occupancy at coordinate $\mathbf{z}$ of the volumetric space can be defined as

$$\mathbf{p}_t[\mathbf{z}] \in \{0, 1\} \tag{3}$$

and we express the predicted probability of coordinate $\mathbf{z}$ being occupied as $\tilde{\mathbf{p}}_t[\mathbf{z}]$, which is obtained using a multi-layer perceptron (MLP) based occlusion network f_{occ}

$$\tilde{\mathbf{p}}_t[\mathbf{z}] = f_{occ}(\mathbf{V}_t[\mathbf{z}]) \tag{4}$$

To perform neural volumetric rendering, we first randomly sample a total of N points $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ along each camera ray as in [62]. However, instead of performing volumetric rendering on RGB values, we render the feature of a given ray $\mathbf{x}$ instead, which can be described as

$$\tilde{\mathbf{V}}_{t,\mathbf{x}} = \sum_{i=1}^{N} \prod_{j=1}^{i-1} (1 - \tilde{\mathbf{p}}_t[\mathbf{x}_j]) \tilde{\mathbf{p}}_t[\mathbf{x}_i] \mathbf{V}_t[\mathbf{x}_i] \tag{5}$$

Finally, the RGB value $\mathbf{c}_{t,\mathbf{x}} \in \mathbb{R}^3$ of the pixel corresponding to ray $\mathbf{x}$ can be obtained through a MLP based reconstruction network f_{recon}

$$\mathbf{c}_{t,\mathbf{x}} = f_{recon}(\tilde{\mathbf{V}}_{t,\mathbf{x}}) \tag{6}$$

Self-supervised Scene Learning: Our approach first trains a voxel-based deformable transformer encoder through self-supervision from a sequence of raw sensor observations, as shown in Fig. 1. Specifically, the transformer-based spatial encoder takes as input the current and T_p frames of past multi-view RGB images $\mathcal{I}_t = \{\mathbf{I}_t, \ldots, \mathbf{I}_{t-T_p}\}$

at timestep t and encodes them into volumetric features $\mathcal{V}_t = \{\mathbf{V}_t, \dots, \mathbf{V}_{t-T_p}\}$. To enforce spatial-temporal consistency over sequential observations, we leverage a *Motion Flow Module* (Fig. 1) that estimates motions between the current frame t to each of the T_f future frames. Specifically, we estimate SE(3) transformations $\mathcal{S}_t = \{\mathbf{S}_{t\to t+1}, \dots, \mathbf{S}_{t+T_f-1\to t+T_f}\}$ between every subsequent future frames, as well as volumetric flow fields $\mathcal{M}_t = \{\mathbf{M}_{t\to t+1}, \dots, \mathbf{M}_{t+T_f-1\to t+T_f}\}$, and $\mathbf{M}_{t\to t+1} \in \mathbb{R}^{X\times Y\times Z\times 3}$. Given a predicted motion, the representation of the current frame $\mathbf{V}_t$ can be transformed to a future frame using the estimated motion flow and SE(3) transformations.

$$\hat{\mathbf{V}}_{t+1} = \mathcal{T}(\mathbf{V}_t, \mathbf{M}_{t\to t+1}, \mathbf{S}_{t\to t+1}) \tag{7}$$

Where the transformation operation $\mathcal{T}$ first transforms the input volumetric feature using the predicted flow field and subsequently applying an SE(3) transformation. That is given flow vector $\in \mathbb{R}^3$ from $\mathbf{M}_{t\to t+n}$, a 3×3 rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$ from $\mathbf{S}_{t\to t+n}$, we transform the coordinate $\mathbf{z}$ to $\hat{\mathbf{z}}$ with

$$\hat{\mathbf{z}} = \mathbf{R}(\mathbf{z} + \mathbf{f}) + \mathbf{t} \tag{8}$$

By predicting the per-voxel flow, our approach is able to effectively handle dynamic scenes, i.e., in contrast to approaches only predicting an SE(3) operation on the volumetric features [9, 31, 52].

Motion Flow Module: We propose a motion flow module to enable fine-grained reasoning over spatio-temporal consistency and dynamic objects. In contrast, prior feature alignment based on rigid transforms [52] or ground-truth camera motion methods will fail under dynamic settings. To better accommodate autonomous driving scenes, we therefore propose to predict both $\mathcal{S}_t$, the rigid SE(3) transformation of ego vehicle, and the voxel-based feature motion flow $\mathcal{M}_t$, which is the 3D motion of each cell of the voxel grid. Specifically, the motion flow module takes as input the concatenated deep voxel features of current and past timesteps, and passes them through several simple 3D ConvNets $\mathcal{H}$ and $\mathcal{G}$, such that $\mathbf{M}_{t\to t+n} = \mathcal{H}(\mathcal{V}_t)$, and $\mathbf{S}_{t\to t+n} = \mathcal{G}(\mathcal{V}_t)$. The predicted SE(3) and flow are then used to warp the deep voxel features into future time steps. In this manner, the motion of each cell can be disentangled while effectively handling dynamic objects.

Loss: We compute an image reconstruction $\mathcal{L}_1$ loss between the rendered image and the original images as well as VGG-16 perceptual loss [47] $\mathcal{L}_{perc}$.

$$\mathcal{L}_{recon} = \sum_{n=0}^{N-1} \|\mathbf{I}_{t+n} - \hat{\mathbf{I}}_{t+n}\|_1 + \lambda_{perc}\mathcal{L}_{perc}(\mathbf{I}_{t+n}, \hat{\mathbf{I}}_{t+n}) \tag{9}$$

In order to ensure the learned volumetric feature captures the 3D structure of the scene, we generate the pseudo occupancy label $\{\mathbf{p}_t, \dots, \mathbf{p}_{t+T_f}\}$ by voxelizing the LiDAR point cloud [22, 57, 91]. We apply an occupancy prediction loss $\mathcal{L}_{occ}$ between predicted occupancy and the pseudo occupancy map, consisting of a scene-class affinity loss $\mathcal{L}_{scal}$ [12] and a binary cross entropy (BCE) loss,

$$\mathcal{L}_{BCE} = -\sum_{n=1}^{T_f-1}\sum_{n=1}^{K}[\mathbf{p}^k_{t+n}\log(\hat{\mathbf{p}}^k_{t+n}) + (1-\mathbf{p}^k_{t+n})\log(1-\hat{\mathbf{p}}^k_{t+n})] \tag{10}$$

$$\mathcal{L}_{occ} = \lambda_{scal}\mathcal{L}_{scal} + \lambda_{BCE}\mathcal{L}_{BCE} \tag{11}$$

3.3 Planning with Volumetric Representation

Problem Setting: We consider the task of learning an end-to-end driving agent $f_{\boldsymbol{\theta}}: \mathcal{X} \rightarrow \mathcal{Y}$ that generates a navigational decision in the form of a desired future trajectory relative to the ego-vehicle, i.e., a set of K waypoints $\mathbf{y} \in \mathcal{Y}$ [18, 63], from observations $\mathbf{x} = (\mathbf{I}_{t-1}, \mathbf{I}_t, v, c) \in \mathcal{X}$ of a sequence of camera images from six perspectives $\mathbf{I}_t = \{\mathbf{I}_t^i\}_{i=1}^6 \in \mathbb{R}^{6\times W\times H\times 3}$ at current timestamp t, $\mathbf{I}_{t-1} = \{\mathbf{I}_{t-1}^i\}_{i=1}^6 \in \mathbb{R}^{6\times W\times H\times 3}$ at the prior timestamp $t-1$, ego-vehicle speed $v \in \mathbb{R}$, and a categorical navigational command $c \in \mathbb{N}$ (e.g., turn left, turn right, and forward [24]). In our work, we aim at learning the driving agent $f_{\boldsymbol{\theta}}$ through behavior cloning. Given a set of collected data of observations and expert trajectories, $\mathcal{D} = \{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1}^N$, the agent can be optimized using

$$\underset{\boldsymbol{\theta}}{\text{minimize}}\, \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\left[\mathcal{L}(\mathbf{y}, f_{\boldsymbol{\theta}}(\mathbf{x}))\right] \tag{12}$$

where $\mathcal{L}$ is a suitable loss function, e.g., L_2 loss for waypoint prediction.

Temporal Attention Volumetric Planner: Traditional end-to-end driving methods tackle the problem in BEV space by converting the front view perception into BEV features, which are used for downstream planning task [17, 43, 85]. However, NeMo leverages a volumetric feature representation $\mathbf{V}_t$ for planning, which provides more fine-grained 3D information of the surrounding environment.

As discussed in Sec. 3.2, in self-supervised scene learning, temporal information is used to train the volumetric features. We propose to incorporate the temporal information through a proposed future attention module for planning. To be specific, the multi-view RGB images of current timestep $\mathbf{I}_t$ and previous timestep $\mathbf{I}_{t-1}$ are encoded into volumetric features $\mathbf{V}_t$ and $\mathbf{V}_{t-1}$, which are concatenated and used to estimate the motion flow $\mathbf{M}_{t\rightarrow t+1}$ and SE(3) transformations $\mathbf{S}_{t\rightarrow t+1}$. Then, the volumetric features $\hat{\mathbf{V}}_{t+1}$ at timestep $t+1$ can be calculated based on Eqn. 7. These future volumetric features are supposed to reason about the dynamic changes of both the ego vehicle and surrounding objects. Then we fuse this future information with the current volumetric features through an attention module. The self-attention is used to compute the attention matrix $\mathbf{A} \in \mathbb{R}^{X\times Y\times Z\times D}$

$$\mathbf{A} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V} \tag{13}$$

where $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ matrices are derived from the estimated future volumetric features $\hat{\mathbf{V}}_{t+1}$. The attention matrix is then used to compute the feature embedding $\mathbf{V}_F \in \mathbb{R}^{X\times Y\times Z\times D}$ for waypoints prediction by multiplying with the current volumetric features $\mathbf{V}_t$. The future attention module ensures temporal feature integration and leads to enhanced focus on the future motion of dynamic objects, which is crucial for motion planning. Our experiments in Sec. 4 demonstrate that attention-based future feature fusion is more effective than simple concatenation or the absence of temporal information.

We adopt a transformer-based planning module for waypoint prediction. The module is built with a standard transformer decoder stacked for three layers [43]. To align with the 2D planning problem, we reduce our feature embedding V_F along the Z dimension by first reshaping the feature with shape $X \times Y \times Z \times D$ into the shape $X \times Y \times (Z * D)$ through merging the height and feature channel dimension. It is further reduced through multiple MLP layers into shape $X \times Y \times C$, where $C = 256$. To enable the planner to reason over high-level navigational commands and ego vehicle status, we fuse navigational command embedding and ego speed embedding, along with a learned planning embedding as query, and cross-attend with the feature embedding. We then use MLP layers to regress our final waypoint prediction.

Loss: To fine-tune the model from the first self-supervised scene learning stage in Sec. 3.2, our loss contains three parts. For the waypoints prediction, a L_2 loss is calculated between the predicted waypoints $\hat{\mathbf{y}}$ and the expert demonstration $\mathbf{y}$:

$$\mathcal{L}_{wpts} = \|\hat{\mathbf{y}} - \mathbf{y}\|_2 \tag{14}$$

To ensure the accurate estimation of the motion flow and SE(3) transformation, thus enabling precise estimation of future volumetric features, we retain the image reconstruction loss and occupancy loss for the current and future frames from Eqn. 9 and Eqn. 11. Therefore, the overall loss for the ego motion planning fine-tuning is defined as

$$\mathcal{L}_{plan} = \lambda_{wpts}\mathcal{L}_{wpts} + \lambda_{recon}\mathcal{L}_{recon} + \lambda_{occ}\mathcal{L}_{occ} \tag{15}$$

where λ_{wpts}, λ_{recon} and λ_{occ} are the weights that balances the tasks.

4 Experiments

We conduct experiments on challenging nuScenes dataset [10], consisting of 1000 driving scenes. We follow standard open-loop evaluation [43, 46, 59, 85] and use L_2 displacement error and the collision rate in 1, 2, and 3 seconds to evaluate the model performance. Additionally, we conduct closed-loop evaluation using the CARLA simulator [28] by adopting the Town05 benchmark following previous studies [23, 46]. We follow the standard CARLA metrics and report Driving Score (DS) and Route Completion (RC), where DS is computed based on RC and infraction rates [1]. In this section, we investigate the following aspects of our design:

1. Is the volumetric representation more effective than BEV representation or simple convolutional neural network (CNN) for motion planning?
2. How do the different modules in pre-training impact the final performance?
3. How do we effectively incorporate temporal information in fine-tuning stage for motion planning task?

4.1 Implementation Details

NeMo adopts ResNet-50 [38] as its default image backbone to encode image features before volumetric feature encoding. We set the range of our volumetric representation

to be $50m \times 50m \times 8m$ and each voxel to be the size of $0.5m$. Our setting results in a volumetric feature the size of $200 \times 200 \times 16$. In the first stage of our model training, we use 4 NVIDIA A6000 GPUs with a batch size of one to train all our models. We adopt AdamW with weight decay $1e^{-2}$ and a learning rate of $2e^{-4}$ and train for a total of 10 epochs. We set the coefficients of the loss function to be $\lambda_{scal} = 10$, and $\lambda_{BCE} = 1$. For the second stage of training, we train our models on four NVIDIA A6000 GPUs, and use AdamW to train our model for 10 epochs with similar settings as the first stage. The hyperparameters to balance various tasks in the fine-tuning stage are set to be $\lambda_{wpts} = 1$, $\lambda_{recon} = 0.5$ and $\lambda_{occ} = 0.5$.

4.2 Results

Model Architecture: We first analyze the proposed volumetric planning model architecture on the nuScenes official validation set. As shown in Table 1, we append the same waypoint prediction head [43] to different perception backbones, and train the models on the nuScenes training set. Previous state-of-the-art baselines use off-the-shelf BEV encoder BEVFormer [55] as their perception backbone, which achieves an average L_2 error of 1.03 without any self-supervised pre-training procedure. Our proposed NeMo (Scratch) planner outperforms the BEV planner and achieves an average L_2 error of 1.02 and an average collision rate of 0.59. Pre-training BEVFormer using the occupancy-based supervision and subsequent fine-tuning for the planning task only results in minor performance gains. In contrast, the rich features learned by NeMo are shown to leverage the pre-training process more effectively and result in larger driving performance gains, e.g., L2 error decreases by 17.6%. The collision rate also shows a noticeable reduction, achieving the lowest rate, of 0.30%, among the various model structures and training settings. We can therefore see the benefits of the volumetric feature representation for motion planning task, even before the self-supervised pre-training, motion flow, or temporal attention mechanism have been applied. We now continue to analyze the advantages of volumetric representation pre-training and temporal attention volumetric planner to improve motion planning performance.

Comparison with State-of-the-art Methods: As shown in Table 1, with self-supervised volumetric feature pre-training, temporal attention mechanism for motion planning, and the proposed designed fine-tuning loss, our proposed NeMo method obtains an average L_2 error of 0.84 and an average collision rate of 0.3%, achieving state-of-the-art performance among prior baseline methods. We note that the volumetric feature pre-training procedure reduces the average L_2 error from 1.02 in Table 1 to 1.00 and average collision rate from 0.59 to 0.54, indicating the effectiveness the self-supervised volumetric feature pre-training mechanism. Next, we will present the detailed ablation of each proposed module.

Closed-Loop Evaluation in CARLA: As shown in Table 2, our NeMo model trained from scratch already outperforms the prior state-of-the-art method in terms of the most important metric, DS (by over 9.5% and 12.9% in Town05 Short and Long benchmark, respectively). This finding highlights the advantages of volumetric feature representation in motion planning tasks. Adding the proposed self-supervised pre-training step

Table 1: Comparative Analysis for Open-Loop Evaluation on nuScenes. Comparison of NeMo with prior baseline methods in terms of L_2 error and collision rate (Collision). VAD [46] computes the error by averaging both over samples and time intervals[1], which is different from other baselines. For a fair comparison, we report VAD results from PARA-Drive [77] (VAD*). NeMo outperforms all prior baseline methods both in terms of L2 error and Collision rate.

Method	L2 (m) ↓				Collision (%) ↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
NMP [84]	-	-	2.31	-	-	-	1.92	-
SA-NMP [84]	-	-	2.05	-	-	-	1.59	-
FF [41]	0.55	1.20	2.54	1.43	0.06	0.17	1.07	0.43
EO [49]	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33
ST-P3 [42]	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
UniAD [43]	0.48	0.96	1.65	1.03	0.05	0.17	**0.71**	0.31
BEVFormer [55] w/o Pre-train	0.37	0.95	1.76	1.03	0.13	0.23	1.44	0.60
BEVFormer [55] w/ Pre-train	**0.35**	0.95	1.76	1.02	0.08	0.27	1.15	0.50
VAD* [77]	0.50	1.02	1.68	1.07	0.02	0.28	0.85	0.38
Our Base Model	0.34	0.92	1.73	1.00	0.07	0.31	1.24	0.54
Base w/ Temporal Concat.	0.45	0.88	1.49	0.94	**0.00**	0.19	0.94	0.38
Base w/ Temporal Attn.	0.36	0.92	1.44	0.91	**0.00**	0.19	0.87	0.35
NeMo (Scratch)	**0.35**	0.94	1.77	1.02	0.11	0.30	1.37	0.59
NeMo	0.39	**0.74**	**1.39**	**0.84**	**0.00**	**0.09**	0.82	**0.30**

further results in improvements to the DS, by 7.7% and 24.4%, and RC, by 8.6% and 13.5%, for the short and long route evaluation, respectively. We note that in contrast to baseline models, e.g., VAD [43, 46], **our model does not leverage any privileged segmentation supervision, such as BEV segmentation annotations**.

4.3 Ablation Studies

Effect of Temporal Module in Motion Planner: Table 1 depicts incorporating temporal information through proper attention module and designed training loss in the fine-tuning stage improves the motion planning performance. Specifically, fusing the predicted future volumetric features with the current volumetric features through the proposed temporal attention module significantly reduces the average L_2 error by 9% (i.e., from 1.00 to 0.91) and the average collision rate by 35% (i.e., from 0.54 to 0.35). It is worth noting that incorporating temporal information benefits long-term planning, leading to better performance in 3 seconds, i.e., 1.73 vs. 1.44 in L_2 error and 1.24 vs. 0.87 in collision rate. Combining temporal information by simply concatenating the

[1] The issue has been publicly discussed on Github: https://github.com/hustvl/VAD/issues/33

Table 2: Comparative Analysis for Closed-loop Evaluation on CARLA. Using camera input only in test-time, NeMo achieves state-of-the-art results in closed-loop simulation on the Town05 CARLA benchmark [23, 28, 46]. We note that *NeMo does not leverage BEV segmentation supervision*, often assumed by prior methods [23, 46].

Method	Town05 Short		Town05 Long	
	DS ↑	RC ↑	DS ↑	RC ↑
CILRS [25]	7.47	13.40	3.68	7.19
LBC [18]	30.97	55.01	7.05	32.09
TransFuser [23]	54.52	78.41	33.15	56.36
ST-P3 [42]	55.14	86.74	11.45	83.15
VAD-Base [46]	64.29	87.26	30.31	75.20
NeMo (Scratch)	70.42	83.01	34.23	71.32
NeMo	**75.87**	**90.12**	**42.57**	**80.98**

Table 3: Component Ablations for the Proposed Self-supervised Pre-training Step. We show each component to holistically contribute to the overall performance of the final model, with occupancy-based supervision being the most impactful. IR refers to Image Reconstruction task. OP refers to Occupancy Prediction task. Flow refers to our proposed motion flow module.

Setup ID.	Settings			L2 (m) ↓				Collision (%) ↓			
	IR	OP	Flow	1s	2s	3s	Avg.	1s	2s	3s	Avg.
0	✓	✓		0.37	0.95	1.75	1.02	0.11	0.37	1.25	0.58
1	✓		✓	0.36	0.96	1.80	1.04	0.10	0.41	1.32	0.61
2		✓	✓	**0.34**	**0.91**	**1.70**	**0.98**	0.09	0.35	1.26	0.57
3	✓	✓	✓	**0.34**	0.92	1.73	1.00	**0.07**	**0.31**	**1.24**	**0.54**

predicted future volumetric features instead of the proposed attention module downgrades planning performance, e.g., 0.94 vs. 0.91 in L_2 error and 0.38 vs. 0.35 in collision rate. Moreover, we observe that fine-tuning the motion planner with auxiliary image reconstruction loss and occupancy loss boosts the performance by 7.7% in average L_2 error and 14.3% in average collision rate. The auxiliary losses guarantee the precise estimation of future volumetric features, contributing to an effective fusion of temporal information.

Effect of Each Module in Self-supervised Pre-training: Table 3 studies the effectiveness of each component proposed in the self-supervised volumetric feature learning. In this study, we fine-tune different pre-trained models with a simple waypoint prediction head [43] appended to the volumetric feature output without temporal information and auxiliary losses. First, we note that the occupancy prediction task substantially contributes to performance gain, e.g., 0.61 vs. 0.54 in the average collision rate (experiment setup ID 1-3 in Table 3). Additionally, the motion flow module enables fine-grained rea-

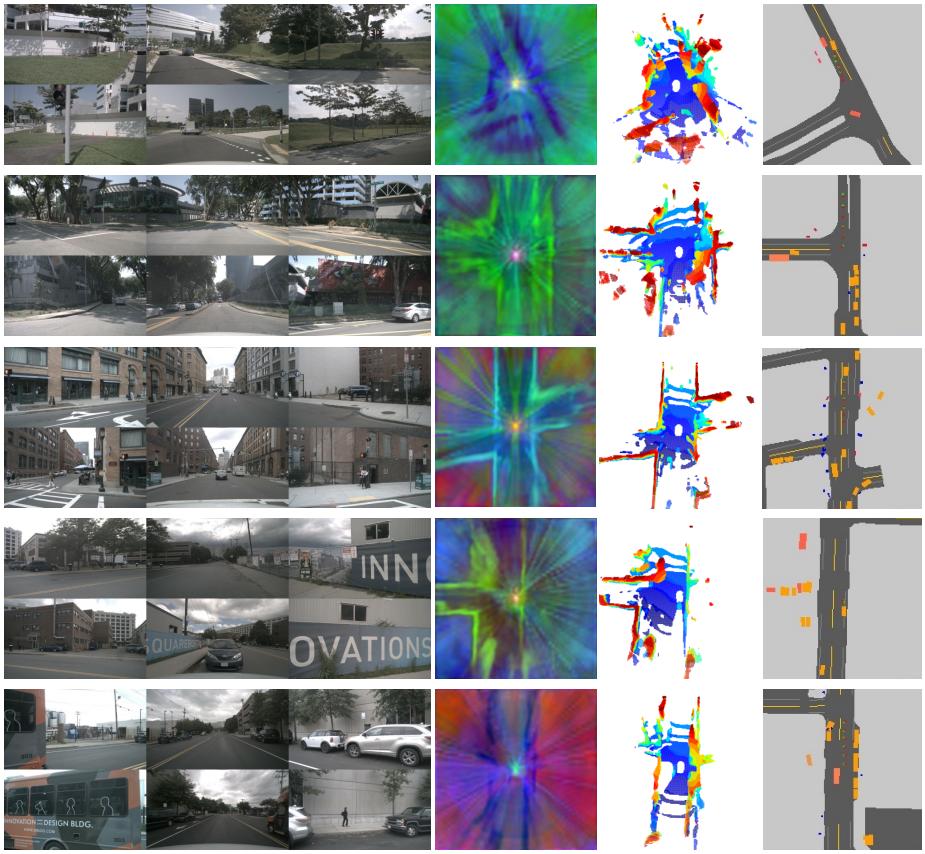

Fig. 3: Qualitative Results on nuScenes. We provide RGB images from six perspectives, volumetric features processed by PCA, occupancy prediction, and ground truth BEV with the planning results. The ground truth waypoints are shown in green and the predicted waypoints are shown in red. Despite the lack of explicit supervision, our method recovers geometry and layout information in the latent space in a self-supervised manner. For instance, in the first row, the vehicle is performing a lane change to the right and the underlying volumetric features highlight road structure which is relevant for the task. We also observe ray artifacts due to the ill-posed rendering-based supervision process. This limitation could be further addressed by future work, e.g., through more sophisticated sampling strategies and auxiliary constraints based on spatial and temporal regularization.

soning over spatio-temporal consistency, leading to 6.9% reduction in average collision rate (ID 0-3). Finally, although the image reconstruction task slightly increases the L_2 error by 2%, the collision rate reduces by 16.9% (ID 2-3). When leveraging all three, NeMo achieves the best motion planning performance with the lowest average L_2 error and collision rate (ID 3).

4.4 Qualitative Results

In Fig. 3, we visualize the learned volumetric features and planning results of NeMo. We provide the raw RGB images from six perspectives and the corresponding ground truth BEV for a better understanding of the scene. To visualize 4D volumetric features $\mathbf{V}_t \in \mathbb{R}^{X \times Y \times Z \times D}$, following previous work [3, 65] we first apply Principal Component Analysis (PCA) to reduce D dimensions to three dimensions, representing the RGB channels. Then we compute the mean along Z dimension, resulting in a feature of size $X \times Y \times 3$. We visualize this feature as a regular RGB image. The visualization shows that NeMo can learn meaningful volumetric features that focus on critical driving-related information i.e., road and objects. For example, in the first row, the ego vehicle is performing lane changing to the right and the visualized PCA processed volumetric feature accurately portrays the road shape which is essential for the planning task. Moreover, the proposed NeMo motion planner is able to leverage the volumetric features and reasonably predict accurate future waypoints. We also observe ray artifacts in PCA processed features due to depth ambiguity in the rendering based supervision process, which could have potential adverse impact on method performance. This limitation could be further addressed by future work through leveraging more intricate and elaborate spatial-temporal constraints in model training. Additional qualitative examples can be found in the supplementary.

5 Conclusion

In this paper, we present NeMo, a volumetric planner through a novel and effective self-supervised feature pre-training. We propose a neural volumetric rendering technique that enables image reconstruction and occupancy prediction as self-supervision. We further design a motion flow module that models the dynamic movement of objects in the scene and enables effective usage of temporal information as additional supervision signals. In the fine-tuning stage, we develop a temporal attention module to fuse the predicted future volumetric features for motion planning. NeMo obtains state-of-the-art results in both closed-loop CARLA evaluation and open-loop evaluation in the nuScenes benchmark, indicating the effectiveness of the proposed framework. Large-scale evaluations in the future can further uncover the benefits of the pre-training stage across driving domains and tasks.

Limitations: While our proposed approach enables more expressive 3D models for visuomotor policies, it introduces certain trade-offs. Given the nature of voxel-based representations, there exists an inherent trade-off between memory and computation efficiency. As memory exhibits a cubit growth as voxel dimensions increase. This is a currently widespread issue with 3D scene modeling and volumetric representations. Nonetheless, recent advances in implementing more efficient volumetric representations (e.g., [61, 69]) may be applicable within our planning-oriented framework. and could be studied in the future.

Acknowledgments: We thank the Red Hat Collaboratory (award #2024-01-RH02) for supporting this research.

References

1. Carla autonomous driving leaderboard. https://leaderboard.carla.org/ (2022)
2. Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: ICCV (2015)
3. Amir, S., Gandelsman, Y., Bagon, S., Dekel, T.: Deep vit features as dense visual descriptors. In: ECCVW (2022)
4. Anderson, P., Chang, A., Chaplot, D.S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M., et al.: On evaluation of embodied navigation agents. arXiv preprint arXiv:1807.06757 (2018)
5. Bain, M., Sammut, C.: A framework for behavioural cloning. In: Machine Intelligence (1996)
6. Bansal, M., Krizhevsky, A., Ogale, A.: Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In: RSS (2019)
7. Behl, A., Chitta, K., Prakash, A., Ohn-Bar, E., Geiger, A.: Label-efficient visual abstractions for autonomous driving. In: IROS (2020)
8. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016)
9. Byravan, A., Fox, D.: SE3-nets: Learning rigid body motion using deep neural networks. In: ICRA (2017)
10. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR (2020)
11. Caine, B., Roelofs, R., Vasudevan, V., Ngiam, J., Chai, Y., Chen, Z., Shlens, J.: Pseudo-labeling for scalable 3d object detection. arXiv preprint arXiv:2103.02093 (2021)
12. Cao, A.Q., de Charette, R.: Monoscene: Monocular 3d semantic scene completion. In: CVPR (2022)
13. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV (2021)
14. Chang, M.F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D., et al.: Argoverse: 3d tracking and forecasting with rich maps. In: CVPR (2019)
15. Chekroun, R., Toromanoff, M., Hornauer, S., Moutarde, F.: Gri: General reinforced imitation and its application to vision-based autonomous driving. arXiv preprint arXiv:2111.08575 (2021)
16. Chen, D., Koltun, V., Krähenbühl, P.: Learning to drive from a world on rails. In: ICCV (2021)
17. Chen, D., Krähenbühl, P.: Learning from all vehicles. In: CVPR (2022)
18. Chen, D., Zhou, B., Koltun, V., Krähenbühl, P.: Learning by cheating. In: CoRL (2020)
19. Chen, L., Sima, C., Li, Y., Zheng, Z., Xu, J., Geng, X., Li, H., He, C., Shi, J., Qiao, Y., Yan, J.: Persformer: 3d lane detection via perspective transformer and the openlane benchmark. In: ECCV (2022)
20. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML (2020)
21. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
22. Cheng, R., Agia, C., Ren, Y., Li, X., Bingbing, L.: S3cnet: A sparse semantic scene completion network for lidar point clouds. In: CoRL (2021)
23. Chitta, K., Prakash, A., Jaeger, B., Yu, Z., Renz, K., Geiger, A.: Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. PAMI (2022)

24. Codevilla, F., Miiller, M., López, A., Koltun, V., Dosovitskiy, A.: End-to-end driving via conditional imitation learning. In: ICRA (2018)
25. Codevilla, F., Santana, E., López, A.M., Gaidon, A.: Exploring the limitations of behavior cloning for autonomous driving. In: ICCV (2019)
26. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
27. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
28. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: CoRL (2017)
29. Ekstrom, A.D., Isham, E.A.: Human spatial navigation: Representations across dimensions and scales. Current opinion in behavioral sciences **17**, 84–89 (2017)
30. Finkelstein, A., Las, L., Ulanovsky, N.: 3D maps and compasses in the brain. Annual Review of Neuroscience (2016)
31. Fu, Y., Misra, I., Wang, X.: Mononerf: Learning generalizable nerfs from monocular videos without camera poses. In: ICML (2022)
32. Gkioxari, G., Ravi, N., Johnson, J.: Learning 3d object shape and layout without 3d supervision. In: CVPR (2022)
33. Gupta, S., Davidson, J., Levine, S., Sukthankar, R., Malik, J.: Cognitive mapping and planning for visual navigation. In: CVPR (2017)
34. Ha, D., Schmidhuber, J.: Recurrent world models facilitate policy evolution. In: NeurIPS (2018)
35. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: CVPR (2022)
36. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. arXiv preprint arXiv:1911.05722 (2019)
37. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
38. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
39. Herweg, N.A., Kahana, M.J.: Spatial representations in the human brain. Frontiers in human neuroscience (2018)
40. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: Octomap: An efficient probabilistic 3d mapping framework based on octrees. Autonomous Robots (2013)
41. Hu, P., Huang, A., Dolan, J., Held, D., Ramanan, D.: Safe local motion planning with self-supervised freespace forecasting. In: CVPR (2021)
42. Hu, S., Chen, L., Wu, P., Li, H., Yan, J., Tao, D.: ST-P3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In: ECCV (2022)
43. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: CVPR (2023)
44. Jaeger, B., Chitta, K., Geiger, A.: Hidden biases of end-to-end driving models. arXiv preprint arXiv:2306.07957 (2023)
45. Jeffery, K.J., Jovalekic, A., Verriotis, M., Hayman, R.: Navigating in a three-dimensional world. Behavioral and Brain Sciences (2013)
46. Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: VAD: Vectorized scene representation for efficient autonomous driving. In: ICCV (2023)
47. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV (2016)
48. Kendall, A., Hawke, J., Janz, D., Mazur, P., Reda, D., Allen, J.M., Lam, V.D., Bewley, A., Shah, A.: Learning to drive in a day. In: ICRA (2019)

49. Khurana, T., Hu, P., Dave, A., Ziglar, J., Held, D., Ramanan, D.: Differentiable raycasting for self-supervised occupancy forecasting. In: ECCV (2022)
50. Lai, L., Ohn-Bar, E., Arora, S., Yi, J.S.K.: Uncertainty-guided never-ending learning to drive. In: CVPR (2024)
51. Lai, L., Shangguan, Z., Zhang, J., Ohn-Bar, E.: XVO: Generalized visual odometry via cross-modal self-training. In: ICCV (2023)
52. Lai, Z., Liu, S., Efros, A.A., Wang, X.: Video autoencoder: self-supervised disentanglement of 3d structure and motion. In: ICCV (2021)
53. LeCun, Y.: A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. Open Review (2022)
54. Li, Y., Yu, Z., Choy, C., Xiao, C., Alvarez, J.M., Fidler, S., Feng, C., Anandkumar, A.: Voxformer: Sparse voxel transformer for camera-based 3D semantic scene completion. In: CVPR (2023)
55. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: ECCV (2022)
56. Liao, Y., Xie, J., Geiger, A.: KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. PAMI (2022)
57. Luo, C., Yang, X., Yuille, A.: Self-supervised pillar motion learning for autonomous driving. In: CVPR (2021)
58. Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., Van Der Maaten, L.: Exploring the limits of weakly supervised pretraining. In: ECCV (2018)
59. Mao, J., Qian, Y., Zhao, H., Wang, Y.: Gpt-driver: Learning to drive with gpt. arXiv preprint arXiv:2310.01415 (2023)
60. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: CVPR (2015)
61. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: CVPR (2019)
62. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
63. Müller, M., Dosovitskiy, A., Ghanem, B., Koltun, V.: Driving policy transfer via modularity and abstraction. arXiv preprint arXiv:1804.09364 (2018)
64. Ohn-Bar, E., Prakash, A., Behl, A., Chitta, K., Geiger, A.: Learning situational driving. In: CVPR (2020)
65. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
66. Pathak, D., Shentu, Y., Chen, D., Agrawal, P., Darrell, T., Levine, S., Malik, J.: Learning instance segmentation by interaction. In: CVPRW (2018)
67. Pomerleau, D.A.: ALVINN: An autonomous land vehicle in a neural network. In: NeurIPS (1989)
68. Qi, W., Mullapudi, R.T., Gupta, S., Ramanan, D.: Learning to move with affordance maps. arXiv preprint arXiv:2001.02364 (2020)
69. Riegler, G., Osman Ulusoy, A., Geiger, A.: Octnet: Learning deep 3d representations at high resolutions. In: CVPR (2017)
70. Spelke, E.S., Lee, S.A.: Core systems of geometry in animal minds. Philosophical Transactions of the Royal Society B: Biological Sciences (2012)
71. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: ICCV (2017)
72. Tian, X., Jiang, T., Yun, L., Mao, Y., Yang, H., Wang, Y., Wang, Y., Zhao, H.: Occ3d: A large-scale 3D occupancy prediction benchmark for autonomous driving. In: NeurIPS (2024)

73. Tolman, E.C.: Cognitive maps in rats and men. Psychological review **55**(4), 189 (1948)
74. Toromanoff, M., Wirbel, E., Moutarde, F.: End-to-end model-free reinforcement learning for urban driving using implicit affordances. In: CVPR (2020)
75. Wang, D., Devin, C., Cai, Q.Z., Krähenbühl, P., Darrell, T.: Monocular plan view networks for autonomous driving. In: IROS (2019)
76. Wei, C., Fan, H., Xie, S., Wu, C.Y., Yuille, A., Feichtenhofer, C.: Masked feature prediction for self-supervised visual pre-training. In: CVPR (2022)
77. Weng, X., Ivanovic, B., Wang, Y., Wang, Y., Pavone, M.: Para-drive: Parallelized architecture for real-time autonomous driving. In: CVPR (2024)
78. Wu, P., Chen, L., Li, H., Jia, X., Yan, J., Qiao, Y.: Policy pre-training for end-to-end autonomous driving via self-supervised geometric modeling. In: ICLR (2023)
79. Wu, P., Jia, X., Chen, L., Yan, J., Li, H., Qiao, Y.: Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. arXiv preprint arXiv:2206.08129 (2022)
80. Wu, S., Jakab, T., Rupprecht, C., Vedaldi, A.: Dove: Learning deformable 3d objects by watching videos. arXiv preprint arXiv:2107.10844 (2021)
81. Yalniz, I.Z., Jégou, H., Chen, K., Paluri, M., Mahajan, D.: Billion-scale semi-supervised learning for image classification. arXiv preprint arXiv:1905.00546 (2019)
82. Yang, C., Chen, Y., Tian, H., Tao, C., Zhu, X., Zhang, Z., Huang, G., Li, H., Qiao, Y., Lu, L., et al.: BEVFormer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In: CVPR (2023)
83. Yang, Z., Chen, L., Sun, Y., Li, H.: Visual point cloud forecasting enables scalable autonomous driving. In: CVPR (2024)
84. Zeng, W., Luo, W., Suo, S., Sadat, A., Yang, B., Casas, S., Urtasun, R.: End-to-end interpretable neural motion planner. In: CVPR (2019)
85. Zhang, J., Huang, Z., Ohn-Bar, E.: Coaching a teachable student. In: CVPR (2023)
86. Zhang, J., Huang, Z., Ray, A., Ohn-Bar, E.: Feedback-guided autonomous driving. In: CVPR (2024)
87. Zhang, J., Ohn-Bar, E.: Learning by watching. In: CVPR (2021)
88. Zhang, J., Zhu, R., Ohn-Bar, E.: SelfD: self-learning large-scale driving policies from the web. In: CVPR (2022)
89. Zhang, Z., Liniger, A., Dai, D., Yu, F., Van Gool, L.: End-to-end urban driving by imitating a reinforcement learning coach. In: ICCV (2021)
90. Zhou, B., Krähenbühl, P.: Cross-view transformers for real-time map-view semantic segmentation. In: CVPR (2022)
91. Zhou, Y., Sun, P., Zhang, Y., Anguelov, D., Gao, J., Ouyang, T., Guo, J., Ngiam, J., Vasudevan, V.: End-to-end multi-view fusion for 3d object detection in lidar point clouds. In: CoRL (2020)
92. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable transformers for end-to-end object detection. In: ICLR (2021)