2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

TransDiffuser: Diverse Trajectory Generation with Decorrelated Multi-modal Representation for End-to-end Autonomous Driving

Xuefeng Jiang[1,2], Yuan Ma[1,3], Pengxiang Li[1,3], Leimeng Xu[1], Xin Wen[1], Kun Zhan[1†], Zhongpu Xia[1], Peng Jia[1], Xianpeng Lang[1], Sheng Sun[2†]

***Abstract*— In recent years, diffusion models have demonstrated remarkable potential across diverse domains, from vision generation to language modeling. Transferring its generative capabilities to modern end-to-end autonomous driving systems has also emerged as a promising direction. However, existing diffusion-based trajectory generative models often exhibit mode collapse where different random noises converge to similar trajectories after the denoising process. Therefore, state-of-the-art models often rely on anchored trajectories from predefined trajectory vocabulary or scene priors in the training set to mitigate collapse and enrich the diversity of generated trajectories, but such inductive bias are not available in real-world deployment, which can be challenged when generalizing to unseen scenarios. In this work, we investigate the possibility of effectively tackling the mode collapse challenge without the assumption of predefined trajectory vocabulary or pre-computed scene priors. Specifically, we propose TransDiffuser, an encoder-decoder based generative trajectory planning model, where the encoded scene information and motion states serve as the multi-modal conditional input of the denoising decoder. Different from existing approaches, we exploit a simple yet effective multi-modal representation decorrelation optimization mechanism during the denoising process to enrich the latent representation space which better guides the downstream generation. Without any predefined trajectory anchors or pre-computed scene priors, TransDiffuser achieves the PDMS of 94.9 on the closed-loop planning-oriented benchmark NAVSIM, surpassing previous state-of-the-art methods. Qualitative evaluation further showcases TransDiffuser generates more diverse and plausible trajectories which explore more drivable area.**

I. Introduction

Recently, substantial advancements have been attained across a diverse range of autonomous driving tasks including real-time localization [1] and scene perception [2]. Among them, planning-oriented autonomous driving [2], [3] has gained widespread attention from both academia and industry for its potential in improving traffic safety and efficiency. Early planning approach usually adopts a sequential process of perception, prediction, and subsequent planning, which causes information loss and cascading latency [4]. Over the past few years, developing fully end-to-end planning-oriented autonomous driving systems [3] has emerged as a key research direction. Taking raw sensor data as input, an end-to-end driving model is expected to directly output an optimal trajectory for guiding future motion planning. Early works [3], [5] aim to generate a single plausible trajectory in an auto-regressive manner by simply imitating annotated human expert driving behaviors in the training dataset.

However, this data-driven paradigm can be not fully capable to generalize to unseen and real-world scenarios. Considering the existence of diverse yet complex driving scenarios and different feasible driving styles [6], there is rarely single feasible trajectory [7]. Recent attempts [8], [9], [7] have increasingly focused on generating multi-mode trajectories as multiple possibly feasible candidates. To generate multi-mode trajectories from the continuous action space, one research line, represented by Hydra-MDP series [9], [10], [11], simplifies this challenge into selecting feasible candidates from a fixed planning vocabulary to discretize the action space. Another emerging research line [7], [12], [6] aims to transfer the success of diffusion models [13], [14] to generate multi-mode trajectories, using scene and motion information as conditional input to produce multi-mode trajectories as candidates. GoalFlow [7] imposes a constraint on the trajectory generation process with scene priors, which is achieved by establishing a dense vocabulary of goal points. DiffusionDrive [12] further highlights the challenge of *mode collapse*, wherein generated trajectories lack diversity as different noise inputs tend to converge to similar trajectory distribution after denoising. It partitions the Gaussian distribution into multiple sub-Gaussian distributions centered around prior anchor trajectories for initialization. Notably, both previous research lines often require the pre-definition of trajectory vocabulary for anchor trajectories or pre-computation of scene priors. This introduces inductive bias which can face challenges for unseen scenarios.

Following the intuition of the above Diffusion based approach, we propose TransDiffuser, an encoder-decoder based multi-mode trajectory generation model. We utilize the frozen Transfuser backbone [5] to encode the scene perception information from front-viewed cameras and LiDAR. The scene information and motion information of the current ego vehicle is then encoded as the conditional input of the Diffusion based denoising decoder. Unlike previous works, we identify another underlying bottleneck that leads to *mode collapse* in generated trajectories: The under-utilization of the encoded multi-modal representation from the conditional input of different modalities. To explore the possibility of generating diverse and feasible trajectories without any anchors or scene priors, inspired by recent advance in self-supervised representation learning, we exploit a computation-efficient yet effective plug-and-play multi-modal representation decorrelation optimization mechanism

[1] LiAuto Inc
[2] Institute of Computing Technology, Chinese Academy of Sciences
[3] School of Vehicle and Mobility, Tsinghua University
[†] Corresponding authors.
✉: jiangxuefeng21b@ict.ac.cn, zhankun@lixiang.com

during the denoising process, which aims to better exploit the multi-modal representation space to guide more diverse feasible planning trajectories from the continuous action space. TransDiffuser achieves the new state-of-the-art performance on the planning-oriented NAVSIM benchmark without any explicit guidance like anchor-based trajectories or scene priors, and qualitative analysis further confirms it can generate diverse trajectories which explore more drivable space. To sum up, our contributions are outlined as follows:

- We propose an encoder-decoder based generative trajectory model TransDiffuser. It firstly encodes the scene perception and motion state of the ego vehicle, and then utilizes the encoded information as conditional input of denoising decoder to decode multi-mode diverse yet feasible trajectories.
- Different from existing works that rely on predefined trajectories or pre-computed scene priors, we exploit a computation-efficient multi-modal representation decorrelation mechanism during the denoising process to enhance the diversity of generated trajectories to address the model collapse dilemma.
- We achieve the new state-of-the-art PDMS 94.9 on NAVSIM benchmark without any explicit guidance like predefined anchor trajectories or scene priors. Qualitative analysis showcases TransDiffuser generates diverse yet feasible trajectories which better explore the drivable space.

II. Related Works

A. End-to-end Autonomous Driving

Modern autonomous driving systems [3], [12], [9], [6] increasingly adopt end-to-end learning paradigms to directly map raw sensor data to driving decisions, streamlining the process from perception to action. Herein we review most recent related advances for planning and categorize current methods into three groups as shown in Table I.

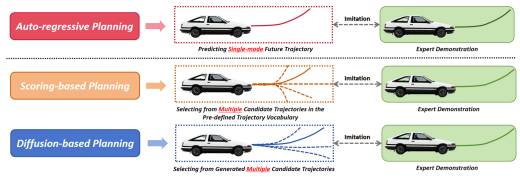

Fig. 1: Main approaches for end-to-end autonomous driving.

Auto-regressive Based Models. This approach is relatively traditional, which typically predict a trajectory via sequential auto-regressive (**AR**) generating. To our best knowledge, UniAD [3] is the pioneering work showcasing the potential of end-to-end autonomous driving by integrating multiple perception tasks to facilitate planning. Transfuser [5] integrates image and LiDAR representations using self-attention and uses GRU to yield the planning trajectory in an auto-regressive paradigm. Its variant LTF is a light-weight image-only version where the LiDAR backbone is replaced by a learnable embedding. PARA-Drive [15] proposes a computation-efficient system which performs mapping, planning, motion prediction and occupancy prediction tasks in parallel.

Scoring Based Models. This approach often generates multi-mode trajectories and selects optimal trajectory with the designed scoring functions. VADv2 [16] is an early attempt to perform multi-mode planning by scoring and sampling from a large fixed vocabulary of anchor trajectories. Centaur [17] employs cluster entropy to measure uncertainty by analyzing the distribution of multiple trajectory candidates generated by the planner. WoTE [18] leverages a latent BEV world model to forecast future BEV states for trajectory evaluation. Hydra-MDP series [9], [10], [11] convert the trajectory generation task into selecting optimal trajectory from predefined trajectory vocabulary. and propose an expert-guided hydra distillation strategy to align the planner with simulation-based metrics. R2SE [19] introduces a reinforced refinement framework with 3D backbone BEVFormer [20] that improves scaled hard case performance.

Diffusion Based Models. One emerging trend aims to transfer the generative capabilities of diffusion models into the end-to-end planning task. Early attempts include GoalFlow [7], DiffusionDrive [12] and TrajHF [6], which will be detailedly discussed in the later Section II-B.

TABLE I: Comparison of existing models. * denotes generating multi-mode candidate trajectories.

Model	Venue	Year	LiDAR	Anchor	Paradigm
Transfuser [5]	ICRA & TPAMI	2023	✓	–	AR
UniAD [3]	CVPR	2023	–	–	AR
PARA-Drive [15]	CVPR	2024	–	–	AR
VADv2 [16]	Arxiv	2024	–	✓	Scoring*
Hydra-MDP [9]	CVPRW	2024	✓	✓	Scoring*
DiffusionDrive [12]	CVPR	2025	✓	✓	Diffusion*
GoalFlow [7]	CVPR	2025	✓	✓	Diffusion*
Hydra-MDP++ [10]	Arxiv	2025	–	✓	Scoring*
Hydra-NeXt [11]	ICCV	2025	–	✓	Scoring*
TrajHF [6]	Arxiv	2025	✓	–	Diffusion*
DIVER [21]	Arxiv	2025	✓	–	Diffusion*
Centaur [17]	Arxiv	2025	✓	✓	Scoring*
WoTE [18]	ICCV	2025	✓	✓	Scoring*
R2SE [19]	TPAMI	2026	–	–	Scoring*
TransDiffuser	ICRA	2026	✓	–	Diffusion*

B. Generative Trajectory Model

Generative models have garnered significant attention due to their remarkable capabilities in producing realistic data across various domains. These models have been successfully applied in numerous tasks, such as image synthesis using Generative Adversarial Network [22] and Variational Autoencoder [23] and text generation with large language models [24]. Diffusion models have been proposed and widely studied in recent years, and demonstrated their powerful generative abilities across vision generation [13], [25], [26], [27] and language generation [28]. A recent shift has emerged in the field of end-to-end autonomous driving, where researchers [12], [7], [6] embark on exploring Diffusion models for planning. Diffusion models, known for their ability to generate high-quality data by iteratively denoising random noise, have shown promise in generating smooth and realistic driving trajectories. However, pointed by DiffusionDrive [12], one bottleneck lies in that diffusion based trajectory generative models showcases the less diversity of the decoded denoised trajectories. *This issue is often called by mode collapse*. It proposes truncated diffusion policy that begins the denoising process from an anchored gaussian distribution instead of

a standard Gaussian distribution. GoalFlow [7] used a goal point vocabulary to mitigate the *mode collapse* issue to assist the efficient trajectory generation. TrajHF [6] pioneers to introduce preference optimization on their private dataset to align the generative trajectory model. Different from above works, we aim to mitigate the *mode collapse* issue via improving the fused intermediate multi-modal representations, which assists to increase the diversity of final decoded trajectories. Notably, we do not utilize any form of guidance like anchors or scene priors, allowing for more flexible and diverse planning.

C. Representation Learning

Representation learning has shown its potential in many deep learning applications like self-supervised learning [29], supervised classification [30], [31], [32], noisy label learning [33], [34], [35], [36] and reinforcement learning [37]. In general, these attempts aim to fully exploit the representation space to learn better intermediate representations for downstream applications. One popular approach is contrastive learning [38] by mining the representation similarities among positive and negative examples. However, this approach generally requires very large training batch (e.g. 4096 or 8192) to fully extract reliable supervision from batch examples, which may be not suitable and computation-efficient for training autonomous driving models. Inspired by the recent advance [37], [33], [30], we optimize the intermediate representations in our task by decoupling the interactions of different representation dimensions. This optimization approach does not rely on the large training batchsize and bring little computation overhead as analyzed in [37], [30], which we will introduce in Section III-D and IV-C.

III. Method

A. Preliminary

As shown in Figure 2, our framework is generally an encoder-decoder model, which consists of two main components: scene encoder and denoising decoder. It takes raw sensor data as input and predicts the future trajectory by accumulating consequent decoded actions. The derived trajectory is represented as a sequence of waypoints (states) $x = \{s_1, s_2, \ldots, s_{\mathcal{T}}\}$, where where $\mathcal{T}$ denotes the trajectory length, and each state s_τ is the location of the $\tau - th$ waypoint in the current driving agent's ego-centric coordinate system. Following [6], each waypoint can connect to its neighbor waypoints by sequentially projecting to action space to mitigate heteroscedasticity along the trajectory sequence timeline. The projection can be recursively expressed as:

$$\hat{x}_\tau, \hat{x}_1 = s_\tau - s_{\tau-1}, s_1 \tag{1}$$

where $\hat{x}_\tau$ represents the agent's action at timestep τ, and τ ranges from 2 to $\mathcal{T}$. This mapping guarantees a reversible connection between the two spaces. As a result, the trajectory can be easily deduced through the accumulation of actions.

B. Scene Encoder and Motion Encoder

We utilize a Scene Encoder (Figure 2(a)) based on the Transfuser backbone [5], pre-trained on Nav-train, to process rich perception information from front-view cameras and LiDAR sensors. This multi-modal approach is crucial as single modalities often lack essential environmental details. Image and LiDAR information are processed through separate backbone networks (dependent branches). We perform multi-stage fusion by connecting features from corresponding stages of both branches via Transformer blocks [39]. This allows for cross-modal attention, leading to enhanced multi-scale representations and better overall integration of sensor data. The final output of Scene Encoder include the image feature F_{img}, LiDAR feature F_{LiDAR} and BEV feature F_{bev}.

To provide essential motion context for predicting future actions, the encoder processes two types of motion information. Firstly, the historical ego trajectory is encoded into an action embedding Emb_{action} using a dedicated Multi-Layer Perceptron (MLP), referred to as the Action Encoder. Secondly, the current ego vehicle status is encoded into an ego status embedding Emb_{ego} by another MLP (Ego Status Encoder). Following [6], these two motion-related embeddings are designed to be utilized within the denoising decoder. The encoded scene and motion feature group $feat = \{F_{bev}, F_{img}, F_{LiDAR}, Emb_{action}.Emb_{ego}\}$ is the conditional input of our denoising decoder.

C. Denoising Decoder

The denoising decoder (Figure 2(b)) is responsible for generating the planned trajectory by iteratively refining an initial noise estimate, conditioned on the encoded scene and motion features derived from the Scene Encoder. It takes the encoded feature group $feat = \{F_{bev}, F_{img}, F_{LiDAR}, Emb_{action}, Emb_{ego}\}$ as conditional input. Different features within this group are sequentially fused via multi-head cross-attention [39]. The decoder then outputs a feasible action from the continuous action space. By periodically accumulating these decoded actions, the final trajectory is formed. To achieve this generation and handle the complexity of driving scenarios, we employ Denoising Diffusion Probabilistic Models (DDPM) [13] as the optimization framework. DDPM involves forward process where noise is gradually added to the ground truth data, and reverse process where the model learns to denoise. Following [6], [12], we focus on reverse denoising process, which transitions from Gaussian noise towards noise-free state x_0. This transition is governed via this equation:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t, feat)) + \sigma_t z \tag{2}$$

where $z \sim \mathcal{N}(0, I)$, $t \in \{1, ..., \mathbf{T}\}$ denotes the noise level timestep, $\mathbf{T}$ is the total number of denoising steps, and ϵ_θ represents our denoising decoder parameterized by θ. The noise prediction ϵ_θ is conditioned on the noisy state x_t, the timestep t, and the feature group $feat$. The parameters $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, and $\sigma_t^2 = \beta_t$ are derived from a predefined noise scheduler β_t. Eq. 2 describes how to estimate

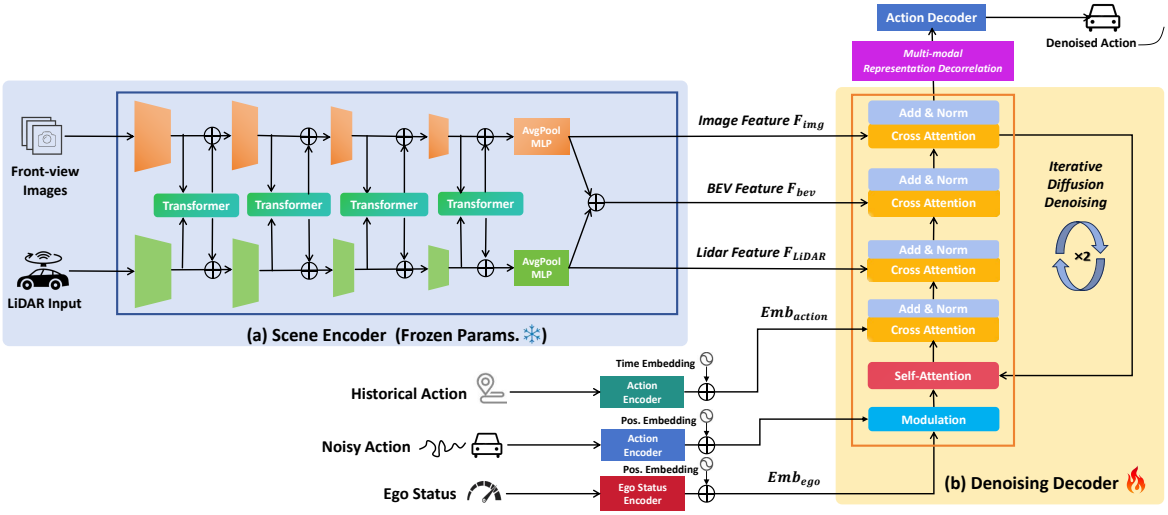

Fig. 2: Overview of the proposed TransDiffuser architecture. We freeze the parameters of the scene encoder.

a slightly less noisy state (x_{t-1}) from the current noisy state (x_t) using the model's noise prediction (ϵ_θ). The decoder is trained to predict the noise added during the forward process. The optimization objective involves minimizing the difference between the actual sampled noise ϵ and predicted noise ϵ_θ, formulated as a gradient descent step on:

$$\nabla_\theta ||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t, feat)||^2 \quad (3)$$

where ϵ is randomly sampled from $\mathcal{N}(0, I)$. Detailed computation of trajectory denoising loss, $\mathcal{L}_{diff}$, using Mean Squared Error (MSE) is outlined in Algorithm 1. In detail, TransDiffuser denoises an 8-step action sequence in parallel during inference which is then cumulatively summed to recover the multiple key waypoints of the complete trajectory.

Algorithm 1: The computation process of $\mathcal{L}_{diff}$

Input: Trajectory GT $label$, Scene feature $feat$, Number of denoising timesteps $\mathbf{T}$
Output: Trajectory loss $\mathcal{L}_{diff}$

// Generate random Gaussian noise
$noise \leftarrow \text{Gaussian}(label.shape)$
// Pick a random time step
$step \leftarrow \text{Rand}(0, \mathbf{T})$
// Apply noise in the forward process
$noisyTarget \leftarrow \text{AddNoise}(label, noise, step)$
// Denoising process
$pred \leftarrow \text{DenoisingDecoder}(noisyTarget, step, feat)$
$\mathcal{L}_{diff} \leftarrow \text{MSE}(pred, noise)$
return $\mathcal{L}_{diff}$

During the inference process, the model leverages this learned denoising process to generate a pool of diverse candidate trajectories starting from pure Gaussian noise. In our implementation, we generate $N = 30$ candidate trajectories considering the efficiency. Subsequently, rejection sampling strategy is employed to filter out dynamically infeasible or unsuitable trajectories following TrajHF [6]. Sensitivity analysis on the candidate number is provided in Section IV-C.

Discussion: GoalFlow [7] typically generates 128 or 256 candidate trajectories, while TrajHF [6] uses 100 candidates prior to their respective selection or filtering steps. As analyzed in Section IV-D, we generate fewer candidates but still maintain effective trajectory quality, our method naturally embodies a degree of inference efficiency.

D. Multi-modal Representation Decorrelation

The final denoised action is decoded by the action decoder as illustrated in Figure 3. We argue planning performance is determined by the quality of the learned multi-modal representations to certain degree. To tackle this information bottleneck [40], we apply the multi-modal representation optimization objective on the fused multi-modal representations, which is illustrated in Figure 3. In each training batch B, we first normalize multi-modal representation matrix $\mathbf{M}$ and then compute its correlation matrix **corr**. This penalty $\mathcal{L}_{reg}$ is designed to decrease non-diagonal entries of the multi-modal correlation matrix **corr**.

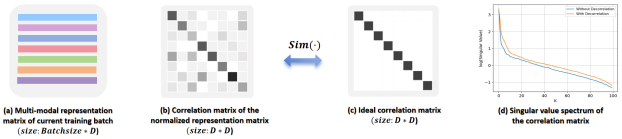

Fig. 3: Illustrations on the multi-modal representation decorrelation. (d) shows top 100 largest the *logarithm* of singular values of the correlation matrix. We observe the decorrelation mechanism helps to better utilize the representation space before action decoding (shown in Figure 2).

Intuitively, $\mathcal{L}_{reg}$ aims to regularize the multi-modal correlation matrix (i.e. Figure 3) to be similar to the diagonal form. This can be achieved by eliminating the unnecessary interactions of different dimensions of representation to increase the similarity (illustrated as $sim(\cdot)$) to identity matrix or diagonal matrix (i.e. Figure 3(c)). As the figure shows, the derived $\mathcal{L}_{rep}$ is a batch-level regularization objective which optimizes the multi-modal representations in each training batch. Since the singular values of the covariance matrix **corr** provide a comprehensive characterization [30] of the distribution of multi-modal representations $\mathbf{M}$, we visualize

the largest singular values shown in Figure 3(d). We also provide the batch-level sensitivity study in Section IV-C.

Algorithm 2: The computation process of L_{reg}

Input: Reshaped multi-modal representation $\mathbf{M}$
Output: L_{reg}

$\mathbf{M}, \sigma_{\mathbf{M}} \leftarrow \mathbf{M} - \text{Mean}(\mathbf{M}, \text{keepdim} = \text{True}), \text{Var}(\mathbf{M}, \text{keepdim} = \text{True})$
//Normalization. ϵ is fixed to $1e^{-8}$ to avoid zero denominator.
$\mathbf{M} \leftarrow \frac{\mathbf{M}}{\sqrt{\epsilon + \sigma_{\mathbf{M}}}}$
//Compute the correlation matrix.
$\mathbf{corr} \leftarrow \mathbf{M}^T \cdot \mathbf{M}$
//Extract the non-diagonal elements.
$c\hat{o}rr \leftarrow \text{remove\_diagonal\_elements}(\mathbf{corr})$
$\mathcal{L}_{rep} \leftarrow (c\hat{o}rr^2.mean())/|\mathbf{B}|$
return $\mathcal{L}_{rep}$

We combine the $\mathcal{L}_{diff}$ and multi-modal representation decorrelation loss with trade-off coefficient β:

$$\mathcal{L} = \mathcal{L}_{diff} + \beta \mathcal{L}_{rep}. \tag{4}$$

Discussion: Herein we simply clarify this novelty. Though representation optimization is relatively widely studied in traditional machine learning tasks [41], [33], [32], it is rarely considered in generative models [42], especially for trajectory generation oriented at end-to-end autonomous driving. In the meantime, decorrelation related computation overhead (FLOPs) is less than 1‰ of the forward process and it benefits the generation diversity as shown in Figure 5.

E. Quantification of Mode Collapse

To evaluate the multi-mode property of the generative trajectory models, we follow DiffusionDrive [12] to develop the quantification analysis. As analyzed in previous diffusion based studies [12], [7], different initialized random noises often lead to converged similar trajectories after the denoising process. To quantitatively analyze the phenomenon of *mode collapse*, we formulate the mode diversity score $\mathcal{D}$ based on the mean Intersection over Union (mIoU) between each denoised trajectory and the union of all denoised trajectories:

$$\mathcal{D} = 1 - \frac{1}{N} \sum_{i=1}^{N} \frac{\text{Area}\left(\tau_i \cap \bigcup_{j=1}^{N} \tau_j\right)}{\text{Area}\left(\tau_i \cup \bigcup_{j=1}^{N} \tau_j\right)} \tag{5}$$

where τ_i represents the $i-th$ denoised trajectory, N is the total number of sampled trajectories and τ_j is the union of all denoised trajectories, as introduced in Section III-C.

IV. Experiments

A. Experimental Setup

Dataset. Following previous state-of-the-art works [12], [7], [6], We utilize the well-established planning-oriented NAVSIM dataset [43] using non-reactive simulation and closed-loop evaluations. It builds on the existing OpenScene [44] dataset, a compact version of nuPlan [45] sampled at 2 Hz. Each sample contains camera images from 8 perspectives, fused LiDAR data from 5 sensors, ego status, and annotations for the map and objects. It has two parts: Nav-train and Nav-test, which respectively contain 1192 and 136 scenarios for training/validation and testing, and we only use the training split for training and validation set to guide model selection.

Baseline. We compare the proposed framework against counterparts from three groups discussed in Section II. Following [43], [7], we additionally refer to *Constant Velocity* and *Ego Status MLP* as the lower bound for comparison. *Constant Velocity* model assumes constant speed from the current timestamp for forward movement. *Ego Status MLP* serves as a bind driving agent, which leverages a MLP for trajectory prediction given only the ego vehicle status.

Implementation Details. Our code is built on the PyTorch lightening framework [46] and official NAVSIM toolkit [43]. We take 10 denoising timesteps during the denoising process for both training and inference. The trade-off coefficient β in Eq. 4 is fixed to 0.02. The learning rate is set to $1e-4$, and global batchsize is 256, distributed across 4 NVIDIA® H20 GPUs. Adam optimizer [47] is adopted with OneCycle scheduler. As the task requires, the model outputs 8 waypoints spanning 4 seconds. Total training lasts for 120 epochs.

Metrics. For NAVSIM dataset, we evaluate our models based on the popular close-looped Predictive Driver Model score (PDMS [48]) which quantifies driving capacity by aggregating sub-metrics for multiple objectives such as progress and comfort. It can be formulated as follows:

$$PDMS = NC \times DAC \times TTC \times \frac{(5 \times DDC + 2 \times C + 5 \times EP)}{12}, \tag{6}$$

where the sub-metrics NC (No at-fault Collision), DAC (Drivable Area Compliance), EP (Ego Progress), $Comfort$, DDC (Driving Direction Compliance), and TTC (Time-to-Collision), each represented as a percentage, are composed into a single score [43]. DDC is exempted from calculation due to the practical constraints of the NAVSIM toolkit [12], [43]. Following the previous work [6], the top-1 predicted trajectory for each sample is used for evaluation. To measure the diversity of all generated candidate trajectories for each sample, we refer to [12] for *Diversity* metric $\mathcal{D}$.

B. Main Experiments

Benchmark Performance. As Table II indicates, TransDiffuser achieves the best performance on Nav-test against baseline methods like DiffusionDrive and GoalFlow. Performance of other models are cited from the original papers. Hydra-MDP++ provides a base version and a large version by scaling up the image encoder from ResNet-34 [49] to V2-99 [50]. The most obvious improvement of TransDiffuser lies in Ego Progress metric, which can also be intuitively observed in Figure 4 and 5. We also witness the potential of Diffusion based models. Note that different from previous state-of-the-art methods like GoalFlow and DiffusionDrive, we do not rely on any priors as discussed in Section II.

TABLE II: Performance on NAVSIM. **L**, **V** and **V*** denote LiDAR input, vision input, and video or historical image input.

Method	Modality	Image Encoder	Metrics					
			$NC\uparrow$	$DAC\uparrow$	$EP\uparrow$	$TTC\uparrow$	$Comfort\uparrow$	**PDMS** $\uparrow$
Constant Velocity [43]	-	-	68.0	57.8	19.4	50.0	100	20.6
Ego Status MLP [43]	-	-	93.0	77.3	62.8	83.6	100	65.6
Transfuser [5]	V+L	ResNet-34	97.8	92.6	78.9	92.9	100	83.9
LTF [5]	V	ResNet-34	97.4	92.8	79.0	92.4	100	83.8
UniAD [3]	V+L	ResNet-34	97.8	91.9	78.8	92.9	100	83.4
PARA-Drive [15]	V*+L	ResNet-34	97.9	92.4	79.3	93.0	99.8	84.0
DiffusionDrive [12]	V+L	ResNet-34	98.2	96.2	82.2	94.7	100	88.1
GoalFlow [7]	V+L	ResNet-34	98.4	98.3	85.0	94.6	100	90.3
VADv2 [16]	V+L	ResNet-34	97.2	89.1	76.0	91.6	100	80.9
Hydra-MDP [9]	V+L	ResNet-34	99.1	98.3	85.2	96.6	100	91.3
Hydra-NeXt [11]	V	ResNet-34	98.1	97.7	81.8	94.6	100	88.6
Hydra-MDP++ (Base) [10]	V*	ResNet-34	97.6	96.0	80.4	93.1	100	86.6
WoTE [18]	V+L	ResNet-34	98.5	95.8	80.9	94.4	99.9	87.1
Hydra-MDP++ (Large) [10]	V*	V2-99	98.6	98.6	85.7	95.1	100	91.0
R2SE [19]	V	BEVFormer	99.0	97.9	86.8	96.4	100	91.6
Centaur [17]	V+L	ResNet-34	**99.5**	**98.9**	85.9	**98.0**	100	92.6
TrajHF [6]	V+L	ViT	99.3	97.5	90.4	**98.0**	99.8	94.0
DIVER [21]	V+L	ResNet-34	98.5	96.5	82.6	94.9	100	88.3
TransDiffuser (Ours)	V+L	ResNet-34	99.4	96.5	**94.1**	97.8	99.4	**94.9**

Diversity Improvement. Following the diversity metric in Section III-E, as shown in Table III and IV, we measure the $\mathcal{D}iversity$ metric over the test dataset. We observe the existence of multi-modal representation decorrelation regularization correspondingly improves the diversity of generated candidate trajectories (from 66 to 70) while there still exists a small gap to DiffusionDrive (74) which benefits from additional trajectory vocabulary to initialize the anchored distribution. In general, TransDiffuser can yield relatively diverse trajectories without any additional priors like anchored trajectories or scene priors. We conduct further analysis on how denoising steps and candidate number affect the diversity in Section IV-C and IV-D, which shows that TransDiffuser can achieves better diversity with more denoising timesteps, and it achieves better performance with 10/15 candidates than 20 candidates of DiffusionDrive.

Training Efficiency. Our training performs 120 epochs across four GPUs and consumes within 2 wall-clock hours. TransDiffuser model has 251M parameters while the parameters of the perception encoder are frozen, thus only 62.8M parameters are trainable. We discuss the inference-time efficiency in Section III-C.

C. Sensitivity Study

We study three key hyper-parameters: denoising timesteps $\mathbf{T}$ in Section III-C, batchsize $\mathbf{B}$ in Section III-D and β in Eq.3. As Table III indicates, our method shows robustness over the selection of coefficient β, denoising timesteps $\mathbf{T}$ and batchsize $\mathbf{B}$. We obverse when we scale up the denoising timesteps for diffusion, the diversity metric $\mathcal{D}$ generally increases at the cost of more computation and latency, while small denoising timesteps can also achieve relatively satisfactory performance with higher efficiency. We visualize the decorrelation effect via the singular value decomposition technique on the correlation matrix in Figure 3. Future works can consider more advanced diffusion policies [7], [51].

TABLE III: Experimental results for the sensitivity study.

Component	Value	Metrics						
		NC	DAC	EP	TTC	$Comfort$	**PDMS**	$\mathcal{D}\uparrow$
Timestep $\mathbf{T}$	5	98.5	94.4	92.7	95.8	99.8	92.4	65
	10	99.4	96.5	94.1	97.8	99.4	94.9	70
	20	99.0	96.0	94.6	96.7	99.3	94.3	88
Batchsize $\mathbf{B}$	32	98.8	94.6	92.9	96.1	99.9	92.7	66
	64	99.4	96.5	94.1	97.8	99.4	94.9	70
	128	98.9	95.4	91.7	97.0	99.0	92.9	69
Coefficient β	0	99.0	95.8	94.5	96.8	99.9	94.3	66
	0.02	99.4	96.5	94.1	97.8	99.4	94.9	70
	0.05	99.0	95.9	93.8	97.2	99.3	94.1	69
	0.1	99.0	95.9	93.3	97.3	99.1	94.0	69

D. Exploration Experiments

Herein we discuss some key exploration regarding our TransDiffuser and multi-modal decorrelation optimization. Experimental results are shown in Table IV.

Candidiate number. The number of candidate trajectories is a key factor to performance. Previous methods like GoalFlow and TrajHF require no less than 100 candidate trajectories. With the assistance of anchored trajectory distribution for generation, DiffusionDrive [12] exhibits satisfactory performance by generating 20 candidates. We decrease the candidate number to 10 and 15, and TransDiffuser still shows its robustness on planning while fewer candidates yield the inferior diversity. Yet future works could explore to achieve high planning performance with fewer candidates [19].

Inner or outer decorrelation. In our original design of TransDiffuser, the decorrelation optimization is adopted on the outside of denoising decoder (i.e. outer decorrelation). We also explore to apply this decorrelation optimization mechanism inside the denoising loop within the denoising decoder. The results indicate that this alternative inner decorrelation counterpart shows a slight decreased performance and diversity.

Holistic training. Scene encoder is frozen in TransDiffuser since we observe the slight performance decrease with the consistent diversity when we fully train the holistic model.

Generalization potential. We apply this decorrelation optimization on the auto-regressive based Transfuser, we find the planning performance gets improved. † in Table IV denotes results are based on our re-implementation.

TABLE IV: Performance on the exploration experiments.

Method	Img. Encoder	Anchor	Candidate(s)	$PDMS \uparrow$	$\mathcal{D} \uparrow$
Transfuser (†)	ResNet-34	×	1	78.0	-
Transfuser (†) + Decorr.	ResNet-34	×	1	78.8	-
GoalFlow	ResNet-34	✓	128/256	90.3	N/A
DiffusionDrive	ResNet-34	✓	20	88.1	74
TrajHF	ViT	×	100	94.0	N/A
Ours	ResNet-34	×	10	89.6	56
Ours	ResNet-34	×	15	91.7	63
Ours	ResNet-34	×	20	93.3	67
Ours	ResNet-34	×	30	94.9	70
Ours (Fully-trained)	ResNet-34	×	10	88.7	55
Ours (Fully-trained)	ResNet-34	×	15	90.8	60
Ours (Fully-trained)	ResNet-34	×	20	92.9	66
Ours (Fully-trained)	ResNet-34	×	30	93.5	68
Ours (Inner Decorr.)	ResNet-34	×	10	86.8	56
Ours (Inner Decorr.)	ResNet-34	×	15	89.8	63
Ours (Inner Decorr.)	ResNet-34	×	20	91.6	66
Ours (Inner Decorr.)	ResNet-34	×	30	93.5	69

E. Qualitative Analysis

Comparison with Transfuser. Since TransDiffuser endows the auto-regressive based Transfuser [5] with generation capabilities to some extent, we provide the visualization of representative examples from the BEV perspective. We visualize the single auto-regressive trajectory predicted by Transfuser and the single selected trajectory of our model. For simple traffic scenarios where there are fewer notable objects, TransDiffuser can propose relatively more aggressive feasible planning trajectories. For complicated traffic scenarios where there are numerous notable objects around the ego driving agent, it tends to propose diverse yet feasible planning trajectories on the premise of ensuring safety.

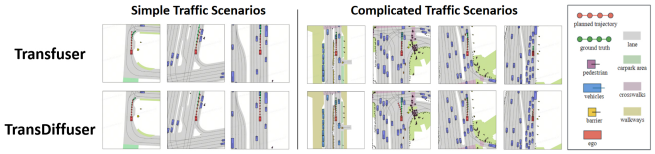

Fig. 4: Visualization of single-mode trajectories.

Comparison with GoalFlow. Since diffusion based counterparts like GoalFlow [7] do not provide specific scene tokens for visualization, we take similar scenes as examples to compare the diversity of generated trajectories in Figure 5. Though we sampled 30 trajectories (fewer than 128/256 trajectories in GoalFlow), we can get more diverse yet feasible trajectories covering more plausible driving space. In contrast, trajectories generated by GoalFlow converge to the similar trajectory distribution given the guidance of goal point priors. By exploiting decorrelation optimization to enrich latent representation space, we can explore more diverse trajectories.

V. Conclusion

In this work, we propose TransDiffuser, an encoder-decoder based generative trajectory model for end-to-end autonomous driving. The encoded scene features and motion features serves as conditional input of the Diffusion-based denoising decoder. We introduce a simple yet effective multi-modal representation decorrelation optimization mechanism to encourage more diverse trajectories from the continuous action space. Experiments on the planning-oriented NAVSIM benchmark demonstrate its superiority in generating high-quality diverse trajectories, without any trajectory anchors or scene priors. For future works, considering more real-world scenarios, the vehicle should generate multiple planning trajectories better aligned with human driver commands or styles, these attempts require reinforcement learning based preference optimization techniques [6], [21] and vision-language-action model architectures [52], [53] to achieve the better balance between effectiveness and safety, which is a crucial concern for deployment. We plan to explore other open-loop datasets and vision-only generative approaches on NAVSIMv2 [54] for open-looped evaluation. For safety-aware applications, we also plan to investigate robust generative trajectory models in the presence of sensor noise (i.e. noisy camera and unstable LiDAR input caused by the bad weather).

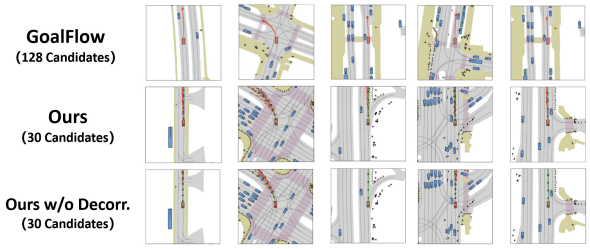

Fig. 5: Visualization on multi-mode diverse trajectories.

Acknowledgment

We appreciate our team's previous work on TrajHF [6]. We also thank our reviewers, Tianyu Li from OpenDrive Lab, Hangjie Mo from Hefei University of Technology for their valuable feedback, as well as Jianwei Ren from Shanghai Qi Zhi Institute, Yida Wang, Yue Wang, Chuan Tang and other researchers and engineers from LiAuto for technical support.

References

[1] X. Jiang, F. Wang, R. Zheng, H. Liu, Y. Huo, J. Peng, L. Tian, and E. Barsoum, "Vips-odom: Visual-inertial odometry tightly-coupled with parking slots for autonomous parking," *arXiv preprint arXiv:2407.05017*, 2024.

[2] X. Tian, J. Gu, B. Li, Y. Liu, Y. Wang, Z. Zhao, K. Zhan, P. Jia, X. Lang, and H. Zhao, "Drivevlm: The convergence of autonomous driving and large vision-language models," in *8th Annual Conference on Robot Learning*.

[3] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, and et. al., "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 17 853–17 862.

[4] A. Sadat, S. Casas, M. Ren, X. Wu, P. Dhawan, and R. Urtasun, *Perceive, Predict, and Plan: Safe Motion Planning Through Interpretable Semantic Representations*, Jan 2020, p. 414–430.

[5] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 11, pp. 12 878–12 895, 2022.

[6] D. Li, J. Ren, Y. Wang, X. Wen, and et. al., "Finetuning generative trajectory model with reinforcement learning from human feedback," *arXiv preprint arXiv:2503.10434*, 2025.

[7] Z. Xing, X. Zhang, Y. Hu, B. Jiang, T. He, Q. Zhang, X. Long, and W. Yin, "Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving," *arXiv preprint arXiv:2503.05689*, 2025.

[8] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Vad: Vectorized scene representation for efficient autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8340–8350.

[9] Z. Li, K. Li, S. Wang, S. Lan, Z. Yu, Y. Ji, Z. Li, Z. Zhu, J. Kautz, Z. Wu, and et. al., "Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation," *arXiv preprint arXiv:2406.06978*, 2024.

[10] K. Li, Z. Li, S. Lan, Y. Xie, Z. Zhang, J. Liu, Z. Wu, Z. Yu, and J. M. Alvarez, "Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation," *arXiv preprint arXiv:2503.12820*, 2025.

[11] Z. Li, S. Wang, S. Lan, Z. Yu, Z. Wu, and J. M. Alvarez, "Hydra-next: Robust closed-loop driving with open-loop training," *arXiv preprint arXiv:2503.12030*, 2025.

[12] B. Liao, S. Chen, H. Yin, B. Jiang, and et. al., "Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving," *arXiv preprint arXiv:2411.15139*, 2024.

[13] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[14] A. Z. Ren, J. Lidard, L. L. Ankile, A. Simeonov, P. Agrawal, A. Majumdar, B. Burchfiel, H. Dai, and M. Simchowitz, "Diffusion policy policy optimization," *arXiv preprint arXiv:2409.00588*, 2024.

[15] X. Weng, B. Ivanovic, Y. Wang, Y. Wang, and M. Pavone, "Para-drive: Parallelized architecture for real-time autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 449–15 458.

[16] S. Chen, B. Jiang, H. Gao, B. Liao, and et. al., "Vadv2: End-to-end vectorized autonomous driving via probabilistic planning," *arXiv preprint arXiv:2402.13243*, 2024.

[17] C. Sima, K. Chitta, Z. Yu, S. Lan, P. Luo, A. Geiger, H. Li, and J. M. Alvarez, "Centaur: Robust end-to-end autonomous driving with test-time training," *arXiv preprint arXiv:2503.11650*, 2025.

[18] Y. Li, Y. Wang, Y. Liu, J. He, L. Fan, and Z. Zhang, "End-to-end driving with online trajectory evaluation via bev world model," *arXiv preprint arXiv:2504.01941*, 2025.

[19] H. Liu, T. Li, H. Yang, L. Chen, and et. al., "Reinforced refinement with self-aware expansion for end-to-end autonomous driving," *arXiv preprint arXiv:2506.09800*, 2025.

[20] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[21] Z. Song, L. Liu, H. Pan, B. Liao, M. Guo, L. Yang, Y. Zhang, S. Xu, C. Jia, and Y. Luo, "Breaking imitation bottlenecks: Reinforced diffusion powers diverse trajectory generation," *arXiv preprint arXiv:2507.04049*, 2025.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[23] D. P. Kingma, M. Welling, and et. al., "Auto-encoding variational bayes," 2013.

[24] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, and et. al., "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[25] J. Li, L. Hu, J. Zhang, T. Zheng, H. Zhang, and D. Wang, "Fair text-to-image diffusion via fair mapping," *CoRR*, vol. abs/2311.17695, 2023.

[26] H. Zhu, D. Tang, J. Liu, M. Lu, J. Zheng, J. Peng, D. Li, Y. Wang, F. Jiang, L. Tian, and et. al., "Dip-go: A diffusion pruner via few-step gradient optimization," *Advances in Neural Information Processing Systems*, vol. 37, pp. 92 581–92 604, 2024.

[27] J. Li, L. Hu, Z. He, J. Zhang, T. Zheng, and D. Wang, "Text guided image editing with automatic concept locating and forgetting," *arXiv preprint arXiv:2405.19708*, 2024.

[28] S. Nie, F. Zhu, Z. You, X. Zhang, and et. al., "Large language diffusion models," *arXiv preprint arXiv:2502.09992*, 2025.

[29] T. Hua, W. Wang, and et. al., "On feature decorrelation in self-supervised learning," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9598–9608.

[30] Y. Shi, J. Liang, and et. al., "Understanding and mitigating dimensional collapse in federated learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 2936–2949, 2024.

[31] X. Lu, P. Li, and X. Jiang, "Fedlf: Adaptive logit adjustment and feature optimization in federated long-tailed learning," in *Asian Conference on Machine Learning*. PMLR, 2025, pp. 303–318.

[32] J. Li and H. Zhang, "Sa-svd: Mitigating bias in face recognition by fair representation learning," in *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 2024, pp. 471–476.

[33] X. Jiang, J. Li, N. Wu, Z. Wu, X. Li, S. Sun, G. Xu, Y. Wang, Q. Li, and M. Liu, "Fnbench: Benchmarking robust federated learning against noisy labels," *Authorea Preprints*, 2024.

[34] Y. Xue, K. Whitecross, and B. Mirzasoleiman, "Investigating why contrastive learning benefits robustness against label noise," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 17–23 Jul 2022, pp. 24 851–24 871.

[35] X. Jiang, P. Li, S. Sun, J. Li, L. Wu, Y. Wang, X. Lu, X. Ma, and M. Liu, "Refining distributed noisy clients: An end-to-end dual optimization framework," *Authorea Preprints*, 2025.

[36] X. Jiang, T. Wen, S. Sun, J. Yuan, H. Liu, P. Li, L. Wu, Y. Wang, and M. Liu, "Representation optimal matching for federated learning with noisy labels in remote sensing," *IEEE Transactions on Mobile Computing*, 2025.

[37] H. Lee, K. Lee, D. Hwang, H. Lee, B. Lee, and J. Choo, "On the importance of feature decorrelation for unsupervised representation learning in reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 18 988–19 009.

[38] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PmLR, 2020, pp. 1597–1607.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[40] R. A. Amjad and B. C. Geiger, "Learning representations for neural network-based classification using the information bottleneck principle," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 9, pp. 2225–2239, 2019.

[41] L. Jing, P. Vincent, Y. LeCun, and Y. Tian, "Understanding dimensional collapse in contrastive self-supervised learning," in *International Conference on Learning Representations*.

[42] R. Wang and K. He, "Diffuse and disperse: Image generation with representation regularization," *arXiv preprint arXiv:2506.09027*, 2025.

[43] D. Dauner, M. Hallgarten, T. Li, X. Weng, and et. al., "Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking," *Advances in Neural Information Processing Systems*, vol. 37, pp. 28 706–28 719, 2024.

[44] S. Peng, K. Genova, C. Jiang, and et. al., "Openscene: 3d scene understanding with open vocabularies," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 815–824.

[45] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, "nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles," *arXiv preprint arXiv:2106.11810*, 2021.

[46] W. A. Falcon, "Pytorch lightning," *GitHub*, vol. 3, 2019.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[48] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," Jun 2023.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[50] Y. Lee, J.-w. Hwang, S. Lee, Y. Bae, and J. Park, "An energy and gpu-computation efficient backbone network for real-time object detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun 2019.

[51] Z. Geng, M. Deng, X. Bai, J. Z. Kolter, and K. He, "Mean flows for one-step generative modeling," *arXiv preprint arXiv:2505.13447*, 2025.

[52] T. Jiang, X. Jiang, Y. Ma, X. Wen, B. Li, K. Zhan, P. Jia, Y. Liu, S. Sun, and X. Lang, "The better you learn, the smarter you prune: Towards efficient vision-language-action models via differentiable token pruning," *arXiv preprint arXiv:2509.12594*, 2025.

[53] J.-J. Hwang, R. Xu, H. Lin, W.-C. Hung, J. Ji, K. Choi, D. Huang, T. He, P. Covington, B. Sapp, and et. al., "Emma: End-to-end multimodal model for autonomous driving," *arXiv preprint arXiv:2410.23262*, 2024.

[54] W. Cao, M. Hallgarten, T. Li, D. Dauner, X. Gu, C. Wang, Y. Miron, M. Aiello, H. Li, I. Gilitschenski, and et. al., "Pseudo-simulation for autonomous driving," *arXiv preprint arXiv:2506.04218*, 2025.