Efficient and Explainable End-to-End Autonomous Driving via Masked Vision-Language-Action Diffusion

Jiaru Zhang, Manav Gagvani, Can Cui, Juntong Peng, Ruqi Zhang, and Ziran Wang

***Abstract*— Large Language Models (LLMs) and Vision-Language Models (VLMs) have emerged as promising candidates for end-to-end autonomous driving. However, these models typically face challenges in inference latency, action precision, and explainability. Existing autoregressive approaches struggle with slow token-by-token generation, while prior diffusion-based planners often rely on verbose, general-purpose language tokens that lack explicit geometric structure. In this work, we propose Masked Vision-Language-Action Diffusion for Autonomous Driving (MVLAD-AD), a novel framework designed to bridge the gap between efficient planning and semantic explainability via a masked vision-language-action diffusion model. Unlike methods that force actions into the language space, we introduce a discrete action tokenization strategy that constructs a compact codebook of kinematically feasible waypoints from real-world driving distributions. Moreover, we propose geometry-aware embedding learning to ensure that embeddings in the latent space approximate physical geometric metrics. Finally, an action-priority decoding strategy is introduced to prioritize trajectory generation. Extensive experiments on nuScenes and derived benchmarks demonstrate that MVLAD-AD achieves superior efficiency and outperforms state-of-the-art autoregressive and diffusion baselines in planning precision, while providing high-fidelity and explainable reasoning. The code is available at `https://github.com/lan-qing/MVLAD-AD`.**

I. Introduction

The paradigm of autonomous driving is shifting from modular pipelines to end-to-end learning systems, where a unified model directly maps raw sensor inputs to driving decisions. However, traditional end-to-end models often function as black boxes, suffering from limited explainability and poor generalization in complex scenarios. Recently, Large Language Models (LLMs) and Vision-Language Models (VLMs) have emerged as promising candidates for autonomous driving, enabling reasoning over complex traffic scenarios and human interaction. By formulating driving as a language modeling problem, these models leverage rich world knowledge from pretraining and improve autonomous driving performance [3], [4], [5].

Despite their promise, current LLM/VLM models for autonomous driving face three main challenges: inference latency, action precision, and explainability. Most existing approaches rely on autoregressive generation. While powerful, as illustrated in Figure 1(A), token-by-token inference is prohibitively slow for the latency-critical nature of autonomous driving. Moreover, processing continuous actions in the language space results in verbose token representations, where describing precise trajectories requires excessive sequence lengths, limiting the inference efficiency of the planning framework. Finally, existing models often struggle to produce plans together with coherent reasoning. Reliance on separate, post-hoc explanation modules often fails to align semantic reasoning with driving actions.

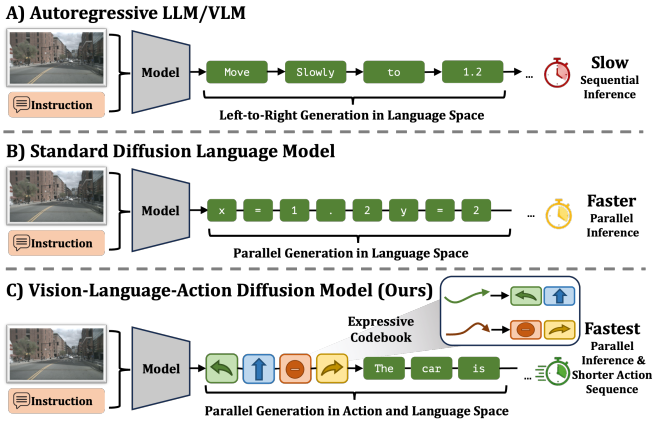

Fig. 1. Motivation. Comparison of paradigms for LLM/VLM-based autonomous driving. (A) Autoregressive VLMs suffer from high latency due to token-by-token sequential generation. (B) Standard diffusion language models enable parallel generation but operate in a verbose language space. (C) Our proposed Vision-Language-Action Diffusion Model incorporates an expressive codebook to map continuous actions into compact discrete tokens. This design enables simultaneous parallel generation in both action and language spaces, significantly reducing sequence length and achieving the fastest inference speed.

Diffusion language models have emerged as a powerful non-autoregressive alternative, enabling parallel decoding to address the inference latency bottleneck [6]. In autonomous driving, ViLaD [7] pioneered this paradigm for efficient planning and achieved substantial improvements in both planning accuracy and inference speed. However, as shown in Figure 1(B), ViLaD relies on verbose language tokens to represent continuous trajectories, introducing representation redundancy. Moreover, it only predicts the driving decisions without any semantic reasoning explanations. These limit both the planning performance and the transparency of current diffusion language model-based planners.

Concurrently, Vision-Language-Action (VLA) models [8], [9] have demonstrated remarkable success by unifying perception, reasoning, and control within a single backbone for embodied AI. By integrating control signals as specialized tokens within the language vocabulary, VLAs facilitate deep

J. Zhang, M. Gagvani and C. Cui contributed equally to this work. The authors are with Purdue University, West Lafayette, IN 47907, USA. Corresponding author: Jiaru Zhang, e-mail: jiaru@purdue.edu. This work used Anvil [1] at Purdue University through allocation CIS251316 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program [2], which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

semantic grounding, enabling the alignment of physical maneuvers with high-level linguistic reasoning. However, standard VLA architectures are predominantly tailored for robotic manipulation tasks, and the best way to leverage VLA modeling for end-to-end autonomous driving remains an open question. Naively discretizing the driving action space results in an explosion of action tokens, rendering the search space intractable and hindering efficient learning.

In this work, we propose Masked Vision-Language-Action Diffusion for Autonomous Driving (MVLAD-AD), a novel framework designed to achieve both efficient planning and semantic explainability. Compared with approaches that force continuous actions into verbose language spaces, we introduce a discrete action tokenization strategy that constructs a compact codebook of kinematically feasible waypoints derived from real-world driving distributions, thereby effectively compressing the action search space. We further propose geometry-aware embedding learning, which aligns the latent embedding space with geometric metrics. These components are integrated into a masked VLA diffusion transformer to model the joint probability of driving actions and linguistic explanations. To resolve the conflict between latency and explainability, we utilize an action-priority decoding strategy that prioritizes trajectory generation during inference. Extensive experiments on the nuScenes and derived reasoning datasets, including Nu-X and nuScenes-QA, demonstrate that MVLAD-AD achieves strong planning performance among VLM-based planners while providing high-fidelity, explainable reasoning.

In summary, our contributions are as follows:

- We propose MVLAD-AD, a novel end-to-end masked VLA diffusion framework that achieves highly efficient end-to-end autonomous driving while retaining semantic reasoning ability.
- We develop discrete action tokenization to map trajectories to compact action tokens, and geometry-aware embedding learning to enforce metric consistency in the embedding space. We further introduce an action-priority decoding strategy to enable low-latency planning.
- We conduct extensive experiments demonstrating that MVLAD-AD outperforms baseline methods on the nuScenes planning benchmark while maintaining superior inference speed and generating coherent, physically aligned linguistic explanations.

II. Related Work

A. Large Language Model-Based End-to-End Autonomous Driving

End-to-end systems are a newly emerging paradigm in autonomous driving that directly maps sensory inputs and ego-vehicle states to planning trajectories and/or low-level control actions [10]. Motivated by the remarkable progress of LLMs, recent studies have explored generative and autoregressive formulations for end-to-end autonomous driving. Chen et al. [11] adopted an autoregressive generation strategy to sequentially predict future scene representations along with corresponding control actions. Following this paradigm, Huang et al. [12] further extended autoregressive modeling to a GPT-style architecture, where future driving scenes are represented and predicted as token sequences.

With the rapid advancement of multimodal foundation models (typically VLMs), their adoption in end-to-end autonomous driving accelerated significantly. Early efforts demonstrated the feasibility of directly incorporating language-centric models into closed-loop driving systems. Shao et al. [13] presented one of the first multimodal LLM-driven end-to-end driving frameworks operating in closed-loop settings . Their system takes navigation goals, multimodal sensory observations, and auxiliary textual instructions as input and directly outputs low-level control commands. Similarly, Wang et al. [14] explored the integration of LLMs into autonomous driving pipelines by aligning language-based decision-making with high-level vehicle control, enabling multimodal inputs and providing interpretable decision rationales.

Beyond direct control generation, a parallel research direction focuses on exploiting the reasoning capabilities of foundation models for decision support and scene understanding. Xu et al. [5] introduced a question–answering-based driving framework that formulates driving decisions as structured queries. Extending this, Sima et al. [15] proposed a graph-based visual question answering (VQA) paradigm, where logically dependent QA pairs capture complex relational reasoning in traffic scenarios. To further enhance explicit reasoning supervision, Wang et al. [16] constructed a chain-of-thought (CoT) driving dataset, annotating both intermediate reasoning steps and final decisions, and proposed a baseline that injects multiple task-specific learnable queries into the reasoning process. More comprehensive multimodal reasoning systems were later introduced by Hwang et al. [17] and Xing et al. [18], which demonstrated strong generalization, robustness, and scalability across diverse driving environments. However, most methods are built upon the autoregressive generation paradigm, resulting in relatively slow inference due to sequential decoding and a left-to-right generation pattern, while primarily modeling short-term dynamics by predicting only the next-step actions or frames.

B. Explainable Autonomous Driving

Explainability has long been recognized as a critical challenge for learning-based autonomous driving systems. As deep neural networks are increasingly adopted for perception, decision-making, and control, their "black box" nature makes it difficult to understand why specific driving actions are produced. This lack of transparency directly affects system verification, safety assurance, trust building, and human–machine collaboration, motivating extensive research on explainable artificial intelligence (XAI) methods for autonomous driving [19], [20].

A broad class of explainability approaches focuses on model-agnostic explanation techniques [21], [22], [23]. These methods aim to explain individual predictions by

estimating feature relevance or attribution without requiring access to the internal structure of the model. In parallel, model-specific techniques have been proposed to analyze internal representations, including gradient-based attribution [24], saliency visualization [25], and attention-based explanations [26].

Within the autonomous driving domain, attention mechanisms have been widely adopted to visualize spatial regions in driving scenes that have a strong influence on control decisions [27]. To further enhance explainability, attention-based driving models have been combined with natural language generation, enabling systems to output both control actions and corresponding textual explanations grounded in visual observations [28]. This line of work was later extended by incorporating linguistic structure and decoding constraints, leading to more coherent and informative textual commentaries for driving behavior [29]. However, attention-based explanations alone have been shown to be insufficient for faithfully reflecting model reasoning [30]. This limitation motivated hybrid explainability strategies that jointly leverage attention weights, encoded features, gradients, and class activation signals to better approximate the underlying decision logic of transformer-based architectures [31].

Recently, LLMs and VLMs introduced a paradigm shift in explainability for autonomous driving, using language as an explicit medium for expressing perception, reasoning, and decision rationales. For example, language-driven driving systems such as DriveLM [15] and LLM-based closed-loop driving frameworks [13], [14], [32], [33], [34] provide textual explanations that articulate what the vehicle observes and why specific actions are taken. Question–answering–based formulations further enable interactive explainability by allowing models to respond to structured queries about driving scenes and decisions [5]. More comprehensive multimodal reasoning frameworks, including EMMA [17] and OpenEMMA [18], demonstrate that language-centric explanations can generalize across diverse driving scenarios while improving robustness and transparency.

III. Methodology

A. Framework Overview

As illustrated in Figure 2, MVLAD-AD formulates end-to-end autonomous driving as a *conditional masked generative modeling* problem. We construct a unified sequence of discrete tokens, $\mathbf{x} = [\mathbf{x}^c; \mathbf{x}^g]$, from heterogeneous modalities. The conditioning part, $\mathbf{x}^c$, aggregates multi-view visual tokens $\mathbf{x}^v$ and textual instruction tokens $\mathbf{x}^i$. The target generation part, $\mathbf{x}^g$, concatenates discrete action tokens $\mathbf{x}^a$ representing the future trajectory and reasoning tokens $\mathbf{x}^r$ explaining the decision. Our objective is to learn the conditional distribution $p_\theta(\mathbf{x}^g \mid \mathbf{x}^c)$ by reversing a parallel masking process [6].

Multimodal Input Encoding. To process the heterogeneous inputs, we employ specialized encoders to project all signals into a shared embedding space, ensuring semantic alignment across modalities:

- **Vision:** Multi-view camera images are encoded by a pre-trained vision encoder and mapped to the transformer's dimension via a learnable MLP Projector.
- **Instruction:** The text instruction is tokenized using a standard text tokenizer and converted into an embedding to condition the model on a specific driving task.
- **Action:** We use our proposed Discrete Action Tokenizer detailed in Section III-B to map continuous waypoints into discrete action tokens, and use our proposed Geometry-Aware Embedding Learning detailed in Section III-C to obtain their embeddings.
- **Reasoning:** Similar to the instruction, the reasoning is tokenized and embedded as text.

Unified Sequence Modeling. Unlike modular pipelines that isolate planning from explanation, MVLAD-AD concatenates these embeddings into a single sequence, $\mathbf{X} = [\mathbf{X}^v; \mathbf{X}^i; \mathbf{X}^a; \mathbf{X}^r]$, where $\mathbf{X}^v$, $\mathbf{X}^i$, $\mathbf{X}^a$, and $\mathbf{X}^r$ correspond to the embeddings of visual features, textual instructions, action tokens, and reasoning tokens, respectively. This unified representation enables the model to leverage bidirectional attention mechanisms and capture the interdependencies among visual observations, text instructions, vehicle actions, and semantic reasoning in a global context.

Masked Generative Transformer. The core of MVLAD-AD is a Transformer-based predictor for masked diffusion generative modeling. As detailed in Section III-D, with a training strategy that jointly learns action and reasoning, the transformer takes the unified sequence as input, with parts of $\mathbf{x}^a$ and $\mathbf{x}^r$ masked, and learns to reconstruct the masked tokens. This architecture effectively models the joint distribution $p_\theta(\mathbf{x}^a, \mathbf{x}^r \mid \mathbf{x}^v, \mathbf{x}^i)$, providing the flexibility to support both efficient parallel planning and reasoning generation, as detailed in Section III-E.

B. Driving Action Tokenization

To bridge the modality gap between continuous trajectory planning and discrete language generation, we propose a Discrete Action Tokenization strategy. In this strategy, the trajectories are modeled as sequences of discrete driving action tokens. Concretely, we represent the trajectory as a series of waypoints in the local ego-coordinate system. Let $\mathbf{w} = (x, y) \in \mathbb{R}^2$ denote a single waypoint representing the vehicle's position relative to the ego-vehicle at a specific future timestamp, and let $\mathcal{D} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$ represent a set of all valid waypoints collected from the real-world driving dataset, where M is the total number of waypoints in the training corpus across all time horizons. Our objective is to construct a compact codebook $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N\}$ consisting of N representative waypoints, such that any continuous waypoint $\mathbf{w} \in \mathcal{D}$ can be approximated by a representative centroid $\mathbf{c} \in \mathcal{C}$ with minimal error. This is formulated as an optimization problem to minimize the within-cluster sum of squares, i.e., $\mathcal{J} = \sum_{i=1}^{M} \min_{\mathbf{c} \in \mathcal{C}} \|\mathbf{w}_i - \mathbf{c}\|_2^2$, where $\|\cdot\|_2$ denotes the Euclidean distance. Utilizing a standard K-Means solver, we obtain the optimal set of spatial centroids $\mathcal{C}$, which serve as our discrete action tokens.

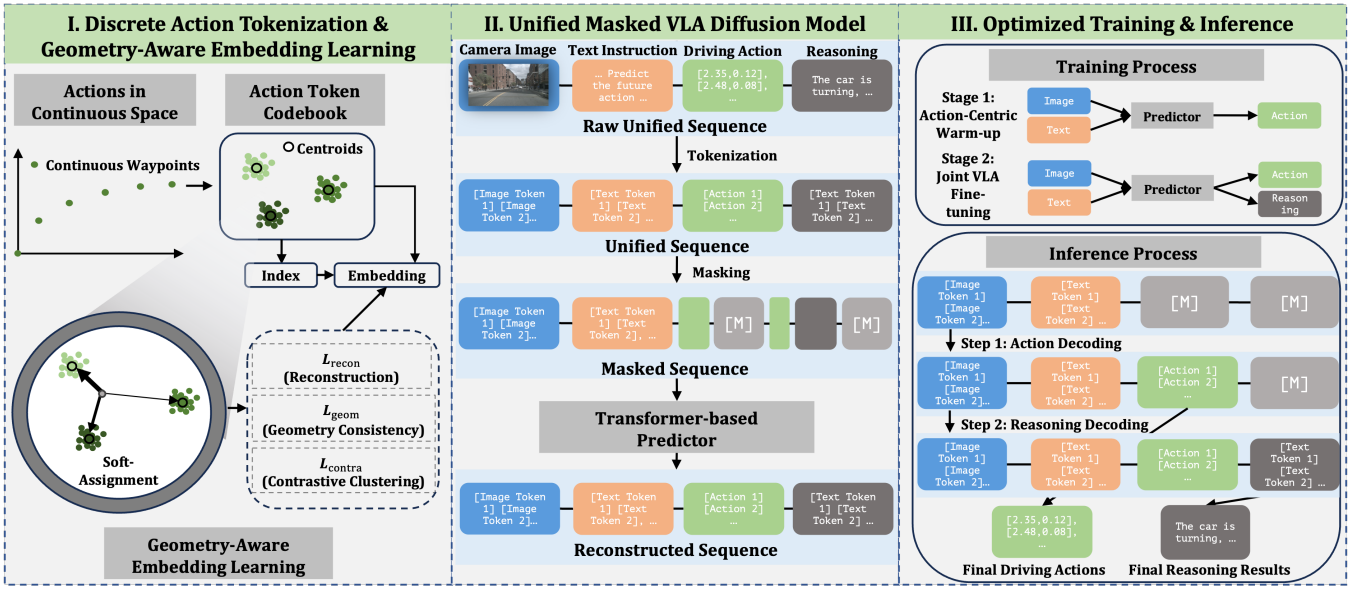

Fig. 2. An overview of MVLAD-AD. (A) Discrete Action Tokenization and Geometry-Aware Embedding Learning: We construct a compact codebook of driving actions from real-world data and learn a geometry-aware embedding space via soft-assignment and geometric consistency objectives. (B) Unified Masked VLA Diffusion: Visual, instruction, action, and reasoning tokens are unified into a single sequence for masked generative modeling. "[M]" indicates masked tokens. (C) Optimized Training & Inference: In training, we employ a two-stage learning strategy. In inference, an action-priority decoding strategy is introduced to prioritize trajectory generation for low latency, ensuring the reasoning explanation is highly faithful to the driving actions.

During training and inference, a quantizer $Q(\cdot)$ is employed to map any continuous predicted waypoint $\hat{\mathbf{w}}$ to the index of its nearest centroid in the codebook:

$$k = \arg\min_{j \in \{1,\dots,N\}} \|\hat{\mathbf{w}} - \mathbf{c}_j\|_2. \tag{1}$$

This discretization transforms the trajectory generation into a sequence of N-way classification task, constraining the output space to physically feasible spatial primitives.

C. Geometry-Aware Embedding Learning

While the action tokenization provides a compact codebook of tokens, treating these tokens as independent categorical indices and using randomly initialized embeddings discards the rich metric information inherent in the trajectory space. To enable the diffusion language model to reason about the physical properties of actions, we propose a pre-training stage to learn a geometry-aware embedding space. Let $\mathbf{E} \in \mathbb{R}^{N \times D}$ denote the learnable embedding matrix for our N action tokens. We aim to optimize $\mathbf{E}$ such that the Euclidean distance in the latent space approximates the geometric distance in the physical space.

Soft-Assignment and Reconstruction. To stabilize the optimization and bridge the gap between continuous inputs and discrete tokens, we employ a temperature-scaled soft-assignment mechanism during training. Given a ground-truth waypoint $\mathbf{w}$, instead of performing a hard lookup, we compute a weighted embedding $\mathbf{z}$ based on the distances to the top-K nearest centroids in the codebook $\mathcal{C}$:

$$\mathbf{z} = \sum_{j \in \mathcal{N}_K(\mathbf{w})} \frac{\exp(-\|\mathbf{w} - \mathbf{c}_j\|_2/\tau)}{\sum_{l \in \mathcal{N}_K(\mathbf{w})} \exp(-\|\mathbf{w} - \mathbf{c}_l\|_2/\tau)} \mathbf{E}_j, \tag{2}$$

where $\mathcal{N}_K(\mathbf{w})$ denotes the set of indices of the K nearest centroids, and τ is a temperature parameter. We then use a lightweight MLP decoder $D_\phi : \mathbb{R}^D \to \mathbb{R}^2$ to reconstruct the original coordinates, minimizing the reconstruction loss $\mathcal{L}_{\text{recon}} = \|D_\phi(\mathbf{z}) - \mathbf{w}\|_2^2$.

Metric Alignment Objectives. To explicitly enforce the geometric structure, we introduce two auxiliary losses:

1) **Geometry Consistency Loss.** We enforce that pairwise distances in the embedding space correlate with physical distances. For a batch of pairs $(\mathbf{w}_i, \mathbf{w}_j)$, we minimize the discrepancy between their normalized distances:

$$\mathcal{L}_{\text{geom}} = \mathbb{E}_{i,j}\left[\left(\frac{\|\mathbf{z}_i - \mathbf{z}_j\|_2}{\bar{d}_z} - \frac{\|\mathbf{w}_i - \mathbf{w}_j\|_2}{\bar{d}_w}\right)^2\right], \tag{3}$$

where $\bar{d}_z$ and $\bar{d}_w$ are the median pairwise distances in the batch, serving as robust scaling factors.

2) **Contrastive Clustering Loss.** We apply a supervised contrastive loss to structure the latent space. For an anchor point i with normalized embedding $\tilde{\mathbf{z}}_i$, let $P(i)$ be the set of other points in the batch assigned to the same centroid index. The loss pulls positive pairs together while pushing apart negatives:

$$\mathcal{L}_{\text{contra}} = \sum_{i \in \mathcal{B}} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\tilde{\mathbf{z}}_i \cdot \tilde{\mathbf{z}}_p/\tau_{\text{con}})}{\sum_{a \in A(i)} \exp(\tilde{\mathbf{z}}_i \cdot \tilde{\mathbf{z}}_a/\tau_{\text{con}})}, \tag{4}$$

where $A(i)$ is the set of all other indices in the batch, and τ_{con} is a contrastive temperature.

Optimization and Curriculum. The final objective is a weighted sum of the three losses. We employ a curriculum learning strategy to ease the transition from continuous to discrete representation. The parameter K decays from 16

to 1 over the course of training. This allows the model to capture the local manifold structure via soft interpolation in the early stages, eventually converging to a hard discrete mapping that yields the embedding matrix $\mathbf{E}$.

D. Training Process

Training a unified VLA model involves learning two distinct capabilities of precise planning and high-level semantic reasoning. Following the standard masked diffusion formulation [6], the base training objective is the negative log-likelihood computed on the masked tokens:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t,\mathbf{x}_0}\left[-\sum_{i\in\mathcal{M}_t}\log p_\theta(x_{0,i}|\mathbf{x}_t)\right], \tag{5}$$

where $\mathcal{M}_t$ denotes the set of masked indices at step t. Since the discrete action tokens introduced in Section III-B represent a new modality with no pre-trained knowledge base, learning their distributions is more challenging than standard language modeling. To address this, we use a two-stage learning strategy.

Stage 1: Action-Centric Warm-up. In the first stage, we isolate the trajectory generation task to initialize the learning of the novel action modality. Concretely, we exclude the reasoning token subsequence $\mathbf{X}^r$ from the input sequence entirely, constructing a shortened sequence $\mathbf{X}_{\text{stage1}} = [\mathbf{X}^v; \mathbf{X}^i; \mathbf{X}^a]$. We mask a portion of the action-token indices and train the model to reconstruct them conditioned solely on visual and instructional contexts. This focused objective forces the model to learn about the physical dynamics and the geometric structure of the action codebook, establishing a robust motion prior without the distraction of the language generation task.

Stage 2: Joint VLA Fine-tuning. In the second stage, we introduce the reasoning task where the unified sequence contains both the action-token subsequence $\mathbf{X}^a$ and the reasoning-token subsequence $\mathbf{X}^r$. We mask both action-token indices and reasoning-token indices, training the model to generate the full output sequence. This stage aligns the physical actions with semantic explanations, leveraging the motion priors established in Stage 1.

E. Inference Process

To balance the conflicting requirements of low-latency planning and semantic explainability, we utilize an action-priority decoding strategy. Standard masked diffusion methods typically employ a global confidence-based sampling schedule, where the tokens with the highest confidence scores across the *entire* sequence are unmasked first. However, this unconstrained approach might generate text tokens and action tokens in a mixed order, delaying the availability of actions.

Instead, we enforce a modality-constrained unmasking policy that prioritizes the resolution of the trajectory. Let $\mathbf{x}_t$ denote the sequence at diffusion step t, containing both action-token positions and reasoning-token positions. In each iteration, the transformer predictor outputs the probability distribution $p_\theta(\mathbf{x}_0 \mid \mathbf{x}_t)$ for *all* masked tokens. Although the model predicts the entire sequence simultaneously, we restrict the unmasking candidate set exclusively to the action indices until planning is fully determined. Specifically, we compute the confidence score $u_j = \max_k p_\theta(x_j = k \mid \mathbf{x}_t)$ for all masked action-token positions j. We then select the subset of action tokens with the highest confidence scores to demask, i.e., replace $[\mathtt{M}]$ with the predicted token ID, while keeping all reasoning tokens in the masked state.

This strategy yields two critical benefits. By focusing the decoding budget solely on the N_a action tokens (where N_a is much smaller than the text length) in the initial steps, the trajectory is finalized and ready for execution significantly faster than waiting for the full sequence to converge. On the other hand, once the action tokens are fully unmasked, they serve as fixed, fully-observed conditions for the subsequent generation of reasoning tokens. This implicitly enforces semantic consistency, as the generated explanation is conditioned on a deterministic future plan.

IV. Experiments

We evaluate the proposed MVLAD-AD framework across both efficient planning and linguistic reasoning. To ensure a comprehensive comparison, we compare against a wide range of baselines, including both open-source state-of-the-art models and representative commercial large language models.

Datasets and Metrics. We primarily utilize the nuScenes dataset [35] for planning evaluation. Specifically, we conduct all experiments and baseline comparisons under a single-frame setting for efficiency. Following standard protocols, we report the L2 displacement error at 1 s, 2 s, and 3 s horizons, along with the average L2 error. For the reasoning tasks, we leverage two specialized datasets: Nu-X [36], a dataset focused on explaining driving decisions, and nuScenes-QA [37], a large-scale visual question-answering benchmark for autonomous driving scenarios. For Nu-X, we employ standard natural language generation metrics, including BLEU-4, METEOR, ROUGE-L, and CIDEr, to measure the semantic alignment between generated explanations and ground truths. For nuScenes-QA, following the literature [36], [38], we report accuracy across zero-hop (H0), one-hop (H1), and overall questions (All).

Baselines. We categorize our baselines based on the task and model architecture:

1) Planning Baselines: We compare MVLAD-AD with a series of representative VLM-based end-to-end driving methods. For autoregressive VLMs, we include models fine-tuned on different backbones: LLaVA-1.6-Mistral-7B, Llama-3.2-11B-Vision-Instruct, and Qwen2-VL-7B-Instruct. We also compare against the best models from the DriveLM [15], OpenEMMA [18], and LightEMMA [39] frameworks. For diffusion VLMs, we include ViLaD [7], which serves as the direct baseline for demonstrating the advantages of our joint VLA modeling and discrete action tokenization. Additionally, we evaluate against UniAD [40], a representative end-to-end autonomous driving model. For a fair comparison,

TABLE I

PLANNING ERRORS (m) AT DIFFERENT HORIZONS AND PLANNING FAILURE RATE (FR). DASHES INDICATE THAT THE METHOD DOES NOT REPORT THE CORRESPONDING RESULTS.

Method	1 s (↓)	2 s (↓)	3 s (↓)	Avg (↓)	FR (↓)
LLaVA-1.6	0.91	2.50	3.44	2.28	55.25%
Llama-3.2	0.80	2.31	3.10	2.07	0.06%
Qwen2-VL	1.32	2.94	3.98	2.74	0.03%
OpenEMMA	1.45	3.21	3.76	2.81	16.11%
LightEMMA	-	-	2.90	1.45	**0.00%**
UniAD-Single	-	-	-	1.80	-
DriveLM	-	-	-	1.39	-
ViLaD	0.81	1.93	2.69	1.81	**0.00%**
MVLAD-AD	**0.70**	**1.31**	**2.34**	**1.28**	**0.00%**

the reported UniAD results are derived from its single-frame version, as evaluated in [15].

2) Explanation Baselines: For reasoning tasks, we compare with a series of specialized models, including Hint-AD [36], ALN-P3 [38], and TOD3Cap [41], which are specifically designed for driving reasoning. We also benchmark against some representative closed-source foundation models, GPT-4o and Gemini-1.5, to estimate how our specialized 7B model compares against general models in industry.

Implementation Details. MVLAD-AD is implemented using the PyTorch framework. We initialize our model weights from the pre-trained LLaDA checkpoint [6]. To improve parameter efficiency, we employ Low-Rank Adaptation (LoRA) with a rank of $r = 256$. Training is distributed across 4 NVIDIA H100 GPUs using `bfloat16` precision, with a global batch size of 32. Each training stage consists of 8 epochs, requiring approximately 9 h to complete. For evaluation, all inference experiments are conducted on a single NVIDIA A100 GPU.

A. Planning Comparison

Superiority of Diffusion Language Modeling. First, we compare diffusion-based architectures, MVLAD-AD and ViLaD, against standard autoregressive VLM baselines. As shown in Table I, diffusion-based methods consistently outperform other baselines across all time horizons. Specifically, MVLAD-AD achieves an average L2 error of 1.28 m, significantly reducing the error compared to the baselines. We attribute this gap to the inherent advantage of diffusion modeling for the planning task, where diffusion-based approaches (MVLAD-AD and ViLaD) allow the model to better capture the nature of driving behaviors without the limitations of sequential prediction.

Benefits of VLA Modeling. Within the diffusion framework, MVLAD-AD further surpasses the previous state-of-the-art method ViLaD, reducing the average L2 error from 1.81 m to 1.28 m. This performance gain verifies the effectiveness of the discrete action tokenization. Unlike ViLaD, which operates only in language space, our framework projects the action space into a compact codebook of $N = 256$ representative action tokens. Therefore, MVLAD-AD performs a classification task over a set of driving actions learned from real-world data. This reduces the complexity of the prediction space, allowing the model to lock onto the optimal trajectory with higher precision and stability.

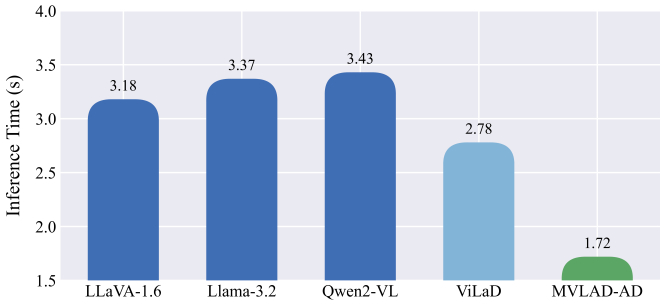

Fig. 3. Planning inference time comparison among autoregressive and diffusion-based methods on a single NVIDIA A100 GPU.

TABLE II

PERFORMANCE COMPARISON ON NU-X

Method	Nu-X			
	CIDEr (↑)	BLEU-4 (↑)	METEOR (↑)	ROUGE-L (↑)
TOD3Cap	14.5	2.45	10.5	23.0
GPT-4o	19.0	3.95	10.3	24.9
Gemini-1.5	17.6	3.43	9.3	23.4
Hint-AD	22.4	4.18	13.2	27.6
ALN-P3	**28.6**	5.59	14.7	35.2
MVLAD-AD	19.5	**13.0**	**36.8**	**37.3**

Systematic Robustness. Reliability is a prerequisite for robotic deployment. We observe that general-purpose VLMs like LLaVA-1.6 suffer from a high failure rate (55.25%), largely due to format hallucinations where the model fails to adhere to the rigid output syntax required for planning. In contrast, MVLAD-AD maintains the zero-failure advantage of LightEMMA and ViLaD, demonstrating a 0.00% failure rate. By mapping the action space to a fixed codebook of valid trajectory centroids and employing a structured decoding process, it structurally guarantees that every generated output corresponds to a valid trajectory. This structural constraint eliminates the risk of format errors inherent in text-based planners, ensuring the system's robustness for downstream execution.

Inference Efficiency. Figure 3 presents the inference latency comparison. Diffusion-based architectures inherently outperform autoregressive baselines by leveraging parallel decoding. MVLAD-AD achieves a further acceleration by introducing a compact VLA modeling scheme. Instead of predicting trajectories via redundant text tokens, our discrete action tokenization compresses complex motion into a minimal sequence of motion primitives. This effectively shortens the sequence length required for planning, significantly reducing the computational workload for the diffusion denoiser. Consequently, MVLAD-AD achieves an inference time of 1.72 s. This represents a 1.6× speedup over the diffusion baseline ViLaD, and a 1.84× speedup over LLaVA-1.6, establishing MVLAD-AD as an efficient solution for end-to-end autonomous driving.

B. Reasoning Comparison

Driving Explanation on Nu-X. Table II presents the quantitative results on the Nu-X dataset. MVLAD-AD demon-

TABLE III
PERFORMANCE COMPARISON ON NUSCENES-QA

Method	nuScenes-QA		
	H0 (↑)	H1 (↑)	All (↑)
TOD3Cap	53.0	45.1	49.0
GPT-4o	42.0	34.7	37.1
Gemini-1.5	40.5	32.9	35.4
Hint-AD	55.4	48.0	50.5
ALN-P3	57.1	50.9	52.9
MVLAD-AD	**58.5**	**54.3**	**55.7**

strates superior performance across most metrics compared to both general-purpose LVLMs, e.g., GPT-4o and Gemini-1.5, and specialized autonomous driving models, including Hint-AD and ALN-P3. Notably, our method achieves a BLEU-4 score of 13.0 and a METEOR score of 36.8, surpassing the previous state-of-the-art method ALN-P3 by a significant margin. Although ALN-P3 yields a higher CIDEr score, MVLAD-AD aligns with the general models like GPT-4o, which also report lower CIDEr scores due to generating more diverse explanations. Furthermore, our substantial lead in BLEU-4 and ROUGE-L indicates that MVLAD-AD generates descriptions that are both semantically richer and more precisely aligned with reference captions in terms of n-gram overlap. It confirms that our model can effectively generate high-quality reasoning texts for complex driving scenarios.

Visual Question Answering on nuScenes-QA. Table III details the evaluation results on the nuScenes-QA benchmark. MVLAD-AD achieves an accuracy of 55.7% on the overall split, consistently outperforming both large-scale commercial models and specialized driving agents. The performance leap indicates that MVLAD-AD captures complex dependencies in driving scenes, enabling the model to answer intricate questions about traffic dynamics with higher precision.

C. Ablation Study

Impact of Action Vocabulary Size N. The size of the discrete codebook N introduces a fundamental trade-off between the planning precision and the learning complexity of the classification task. Because our framework formulates trajectory planning as a discrete token prediction task, the training loss primarily reflects the classification difficulty, whereas the L2 error measures the final planning performance. Theoretically, a larger N reduces the quantization error, i.e., the spatial distance between a continuous waypoint and its nearest centroid, thereby raising the upper bound of reconstruction fidelity. However, an excessive number of tokens increases the complexity of the classification task.

We investigate this trade-off by training MVLAD-AD with vocabulary sizes $N \in \{128, 256, 384\}$, as reported in Table IV. We observe that $N = 256$ strikes the optimal balance, achieving the lowest planning error of 1.28 m. Crucially, increasing N to 384 leads to a performance degradation to 2.76 m with an obvious increase in final training loss from 0.36 to 0.53. This indicates that, despite higher theoretical precision, the model struggles to converge due to the optimization difficulty in distinguishing between dense action tokens. Conversely, reducing N to 128 yields the lowest training loss of 0.32, suggesting an easier classification task, but the planning error rises to 1.73 m. This confirms that further reducing the codebook size creates a quantization bottleneck that limits physical precision, regardless of stable training convergence.

TABLE IV
ABLATION ON ACTION VOCABULARY SIZE N

Number of Action Tokens N	Final Training Loss (↓)	L2 (m) Avg. (↓)
128	0.32	1.73
256 (Default)	0.36	**1.28**
384	0.53	2.76

Effectiveness of Geometry-Aware Embedding Learning. We investigate the impact of geometry-aware embedding learning on the planning metrics of the model. As a comparison, we remove this module and train the model with random embeddings. This causes a substantial performance degradation, increasing the average L2 error from 1.28 m to 2.39 m. This comparison confirms that treating action tokens as independent categorical indices discards useful metric information, making it difficult for the model to learn effective planning. By enforcing geometric consistency, our method ensures that the distance between tokens in the latent space correlates with their physical displacement, enabling the model to generate accurate trajectories.

Action Representation: Waypoints vs. Displacements. We compare modeling trajectories as absolute waypoints versus relative displacements [42]. As shown in Table V, while displacements barely impact planning accuracy (L2 error rises slightly from 1.28 m to 1.30 m), they cause a collapse in reasoning. The displacement model fails to generate coherent explanations, dropping the CIDEr score to 0.08 and reducing BLEU-4. This validates our use of absolute waypoints. Although displacements provide enough local detail for short-term planning, they lack the global spatial context required for semantic alignment. Since diffusion generates in parallel, the model struggles to aggregate isolated displacements to understand the overall maneuver. Without explicit spatial anchors, it cannot map physical actions to high-level concepts, leading to an obvious breakdown in explainability.

TABLE V
ABLATION ON ACTION REPRESENTATION. C: CIDER, B: BLEU-4, M: METEOR, R: ROUGE-L.

Modeling	Planning	Nu-X			
	L2 (m) Avg. (↓)	C (↑)	B (↑)	M (↑)	R (↑)
Displacement	1.30	0.08	5.66	23.1	26.4
Waypoint	**1.28**	**19.5**	**13.0**	**36.8**	**37.3**

V. CONCLUSION

In this work, we introduced MVLAD-AD, a unified framework for end-to-end autonomous driving that balances low-latency, high-precision planning with semantic explainability. Instead of regular language space representations, our

discrete action tokenization strategy successfully transforms continuous trajectory planning into a robust classification task over kinematically feasible motion primitives. Moreover, our geometry-aware embedding bridges the gap between semantic reasoning and physical dynamics, enabling the system to generate high-fidelity trajectories with physically grounded explanations. Our action-priority decoding strategy further reduces the planning latency in inference. Extensive evaluations demonstrate that MVLAD-AD establishes strong performance on nuScenes planning and related language benchmarks, significantly outperforming autoregressive baselines in both planning precision and inference speed, while also providing high-quality explainable reasoning.

REFERENCES

[1] X. C. Song, P. Smith, R. Kalyanam, X. Zhu, E. Adams, K. Colby, P. Finnegan, E. Gough, E. Hillery, R. Irvine, *et al.*, "Anvil - System Architecture and Experiences from Deployment and Early User Operations," in *PEARC*, 2022.

[2] T. J. Boerner, S. Deems, T. R. Furlani, S. L. Knuth, and J. Towns, "ACCESS: Advancing Innovation: NSF's Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support," in *PEARC*, 2023.

[3] J. Mao, Y. Qian, J. Ye, H. Zhao, and Y. Wang, "GPT-Driver: Learning to Drive with GPT," *arXiv preprint arXiv:2310.01415*, 2023.

[4] L. Wen, D. Fu, X. Li, X. Cai, T. MA, P. Cai, M. Dou, B. Shi, L. He, and Y. Qiao, "DiLu: A Knowledge-Driven Approach to Autonomous Driving with Large Language Models," in *ICLR*, 2024.

[5] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K.-Y. K. Wong, Z. Li, and H. Zhao, "DriveGPT4: Interpretable End-to-end Autonomous Driving via Large Language Model," *IEEE Robotics and Automation Letters*, vol. 9, no. 10, 2024.

[6] S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. ZHOU, Y. Lin, J.-R. Wen, and C. Li, "Large Language Diffusion Models," in *NeurIPS*, 2025.

[7] C. Cui, Y. Zhou, J. Peng, S.-Y. Park, Z. Yang, P. Sankaranarayanan, J. Zhang, R. Zhang, and Z. Wang, "ViLaD: A Large Vision Language Diffusion Framework for End-to-End Autonomous Driving," *arXiv preprint arXiv:2508.12603*, 2025.

[8] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, *et al.*, "OpenVLA: An Open-Source Vision-Language-Action Model," in *CoRL*, 2024.

[9] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, *et al.*, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," in *CoRL*, 2023.

[10] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end Autonomous Driving: Challenges and Frontiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, 2024.

[11] Y. Chen, Y. Wang, and Z. Zhang, "DrivingGPT: Unifying Driving World Modeling and Planning with Multi-modal Autoregressive Transformers," in *ICCV*, 2025.

[12] X. Huang, E. M. Wolff, P. Vernaza, T. Phan-Minh, H. Chen, D. S. Hayden, M. Edmonds, B. Pierce, X. Chen, P. E. Jacob, *et al.*, "DriveGPT: Scaling Autoregressive Behavior Models for Driving," *arXiv preprint arXiv:2412.14415*, 2024.

[13] H. Shao, Y. Hu, L. Wang, G. Song, S. L. Waslander, Y. Liu, and H. Li, "LMDrive: Closed-Loop End-to-End Driving with Large Language Models," in *CVPR*, 2024.

[14] E. Cui, W. Wang, Z. Li, J. Xie, H. Zou, H. Deng, G. Luo, L. Lu, X. Zhu, and J. Dai, "DriveMLM: Aligning Multi-Modal Large Language Models with Behavioral Planning States for Autonomous Driving," *Visual Intelligence*, vol. 3, no. 1, p. 22, 2025.

[15] C. Sima, K. Renz, K. Chitta, L. Chen, H. Zhang, C. Xie, J. Beißwenger, P. Luo, A. Geiger, and H. Li, "DriveLM: Driving with Graph Visual Question Answering," in *ECCV*, 2024.

[16] T. Wang, E. Xie, R. Chu, Z. Li, and P. Luo, "DriveCoT: Integrating Chain-of-Thought Reasoning with End-to-End Driving," *arXiv preprint arXiv:2403.16996*, 2024.

[17] J.-J. Hwang, R. Xu, H. Lin, W.-C. Hung, J. Ji, K. Choi, D. Huang, T. He, P. Covington, B. Sapp, Y. Zhou, J. Guo, D. Anguelov, and M. Tan, "EMMA: End-to-End Multimodal Model for Autonomous Driving," *Transactions on Machine Learning Research*, 2025.

[18] S. Xing, C. Qian, Y. Wang, H. Hua, K. Tian, Y. Zhou, and Z. Tu, "OpenEMMA: Open-Source Multimodal Model for End-to-End Autonomous Driving," in *WACV Workshops*, 2025.

[19] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information fusion*, vol. 58, pp. 82–115, 2020.

[20] D. Omeiza, H. Webb, M. Jirotka, and L. Kunze, "Explanations in Autonomous Driving: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 142–10 162, 2021.

[21] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *NIPS*, 2017.

[22] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning Important Features Through Propagating Activation Differences," in *ICML*, 2017.

[23] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," in *SIGKDD*, 2016.

[24] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization," in *ICCV*, 2017.

[25] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," *arXiv preprint arXiv:1312.6034*, 2013.

[26] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," in *ICML*, 2015.

[27] J. Kim and J. Canny, "Interpretable Learning for Self-Driving Cars by Visualizing Causal Attention," in *ICCV*, 2017.

[28] J. Kim, A. Rohrbach, T. Darrell, J. Canny, and Z. Akata, "Textual Explanations for Self-Driving Vehicles," in *ECCV*, 2018.

[29] M. A. Kühn, D. Omeiza, and L. Kunze, "Textual Explanations for Automated Commentary Driving," in *IV*, 2023.

[30] S. Jain and B. C. Wallace, "Attention is not explanation," in *NAACL*, 2019.

[31] Y. Qiang, D. Pan, C. Li, X. Li, R. Jang, and D. Zhu, "AttCAT: Explaining Transformers via Attentive Class Activation Tokens," in *NeurIPS*, 2022.

[32] L. Chen, O. Sinavski, J. Hünermann, A. Karnsund, A. J. Willmott, D. Birch, D. Maund, and J. Shotton, "Driving with LLMs: Fusing Object-Level Vector Modality for Explainable Autonomous Driving," in *ICRA*, 2024.

[33] C. Cui, Z. Yang, Y. Zhou, Y. Ma, J. Lu, L. Li, Y. Chen, J. Panchal, and Z. Wang, "Personalized Autonomous Driving with Large Language Models: Field Experiments," in *ITSC*, 2024.

[34] C. Cui, Y. Ma, S.-Y. Park, Z. Yang, Y. Zhou, P. Liu, J. Lu, J. Peng, J. Zhang, R. Zhang, *et al.*, "LLM4AD: Large Language Models for Autonomous Driving—Concept, Review, Benchmark, Experiments, and Future Trends," *Proceedings of the IEEE*, 2026.

[35] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A Multimodal Dataset for Autonomous Driving," in *CVPR*, 2020.

[36] K. Ding, B. Chen, Y. Su, H.-a. Gao, B. Jin, C. Sima, X. Li, W. Zhang, P. Barsch, H. Li, *et al.*, "Hint-AD: Holistically Aligned Interpretability in End-to-End Autonomous Driving," in *CoRL*, 2024.

[37] T. Qian, J. Chen, L. Zhuo, Y. Jiao, and Y.-G. Jiang, "NuScenes-QA: A Multi-Modal Visual Question Answering Benchmark for Autonomous Driving Scenario," in *AAAI*, 2024.

[38] Y. Ma, B. Yaman, X. Ye, M. Yurt, J. Luo, A. Mallik, Z. Wang, and L. Ren, "ALN-P3: Unified Language Alignment for Perception, Prediction, and Planning in Autonomous Driving," *arXiv preprint arXiv:2505.15158*, 2025.

[39] Z. Qiao, H. Li, Z. Cao, and H. X. Liu, "LightEMMA: Lightweight End-to-End Multimodal Model for Autonomous Driving," *arXiv preprint arXiv:2505.00284*, 2025.

[40] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, *et al.*, "Planning-Oriented Autonomous Driving," in *CVPR*, 2023.

[41] B. Jin, Y. Zheng, P. Li, W. Li, Y. Zheng, S. Hu, X. Liu, J. Zhu, Z. Yan, H. Sun, *et al.*, "TOD3Cap: Towards 3D Dense Captioning in Outdoor Scenes," in *ECCV*, 2024.

[42] W. Tang, J. You, J. Liu, Z. Wang, R. Gan, Z. Huang, F. Wei, and B. Ran, "HERMES: A Holistic End-to-End Risk-Aware Multimodal Embodied System with Vision-Language Models for Long-Tail Autonomous Driving," *arXiv preprint arXiv:2602.00993*, 2026.