ViLaD: A Large Vision Language Diffusion Framework for End-to-End Autonomous Driving

Can Cui, Jiaru Zhang, Yupeng Zhou, Juntong Peng, Sung-Yeon Park, Zichong Yang,
Ruqi Zhang, and Ziran Wang

***Abstract*— End-to-end autonomous driving systems built on Vision Language Models (VLMs) have shown significant promise, yet their reliance on autoregressive architectures introduces some limitations for real-world applications. The sequential, token-by-token generation process of these models results in high inference latency and cannot perform bidirectional reasoning, making them unsuitable for dynamic, safety-critical environments. To overcome these challenges, we introduce ViLaD, a novel Large Vision Language Diffusion (LVLD) framework for end-to-end autonomous driving that represents a paradigm shift. ViLaD leverages a masked diffusion model that enables parallel generation of entire driving decision sequences, significantly reducing computational latency. Moreover, its architecture supports bidirectional reasoning, allowing the model to consider both past and future simultaneously, and supports progressive easy-first generation to iteratively improve decision quality. We conduct comprehensive experiments on the nuScenes dataset, where ViLaD outperforms state-of-the-art autoregressive VLM baselines in both planning accuracy and inference speed, while achieving a near-zero failure rate. Furthermore, we demonstrate the framework's practical viability through a real-world deployment on an autonomous vehicle for an interactive parking task, confirming its effectiveness and soundness for practical applications.**

I. Introduction

Autonomous driving represents one of the most challenging applications of artificial intelligence, requiring real-time perception, reasoning, and decision-making in dynamic, safety-critical environments [1]. As an alternative to traditional module-based pipelines, end-to-end autonomous driving aims to use a single model to directly map raw sensor data to driving commands, which has become a popular autonomous driving paradigm. Recent advances have shown remarkable promise for end-to-end autonomous driving by enabling vehicles to process multimodal sensory inputs and generate human-interpretable driving decisions through Vision Language Models (VLMs) [2], [3], [4]. These approaches leverage the powerful reasoning capabilities of VLMs to bridge the gap between raw sensor data and high-level driving commands, offering better interpretability and flexibility in autonomous driving systems.

However, current VLM-based end-to-end autonomous driving systems rely on autoregressive architectures that generate decisions through sequential next-token prediction. This autoregressive paradigm, while successful for many natural language processing tasks, presents some fundamental limitations when applied to the time-sensitive and spatially complex domain of real-world driving scenarios. These models generate decisions sequentially, token-by-token, which is *computationally inefficient* and *conceptually inflexible*: First, the sequential generation process creates significant inference latency, as each token must be generated one by one, making real-time decision-making challenging for safety-critical scenarios; Second, the left-to-right generation paradigm limits the model's ability to consider global spatial relationships and future trajectory implications simultaneously, potentially leading to suboptimal driving decisions that lack comprehensive scene understanding. In the dynamic and unpredictable environment of the road, this lack of foresight and adaptability may cause a significant challenge to safety and reliability; Moreover, autoregressive VLMs suffer from the "reversal curse" phenomenon [5], where models trained on sequential patterns struggle with tasks requiring bidirectional reasoning or reverse inference. In autonomous driving, this limitation causes difficulty in reasoning backward from desired outcomes (e.g., "To avoid the congested traffic flow, I should exit the highway at the next exit") or integrating future planning constraints into current decisions (e.g., "Stop at the next intersection").

To address these fundamental limitations, we propose a fundamental paradigm shift for end-to-end autonomous driving from autoregressive VLMs to the Large Vision Language Diffusion (LVLD) models [6]. These kinds of models rely on a masked diffusion generation paradigm. Specifically, the text is generated by starting with a fully masked sequence, which is then iteratively filled in with tokens predicted by a model trained to reverse this masking process. Therefore, it offers several advantages in end-to-end autonomous driving:

- **Efficient Parallel Generation**. Unlike sequential token prediction, diffusion models generate the decision sequences simultaneously, reducing inference latency.
- **Bidirectional Reasoning**. The iterative bidirectional generation process enables models to consider both past context and future influences comprehensively.
- **Easy-First Generation Patterns**. The model can adaptively focus on easier aspects of the driving task first, then progressively address more complex aspects.

With the utilization of LVLD models, we introduce **ViLaD** (**Vi**sion **La**nguage **D**iffusion), a novel end-to-end autonomous driving framework that leverages the masked diffusion generation method. This technique trains a neural network model to reconstruct a complete sequence of driving actions from a version where parts of the output are

The authors are with Purdue University, West Lafayette, IN 47907, USA. Corresponding author: J. Zhang, email: jiaru@purdue.edu.

deliberately "masked." During inference, the model begins with a fully masked (except fixed tokens) sequence of decision tokens and iteratively "unmasks" the masked tokens in parallel over several steps to produce the final output. It demonstrates a successful end-to-end implementation of the LVLD autonomous driving system from the benchmark comprehensive experiment to a real-world vehicle deployment, which is the **first of its kind** to the best of our knowledge. Our main contributions are summarized as follows:

- We are the first to adapt LVLD models for end-to-end autonomous driving tasks, establishing a new paradigm that moves beyond autoregressive generation to enable more efficient and comprehensive decision-making in safety-critical driving scenarios.
- We develop and analyze novel inference optimization strategies tailored for diffusion-based driving models, including a Template-Constrained Action Masking (TCAM) method and a Confidence-Guided Easy-First Planning (CGEFP) decoding approach. These techniques significantly improve inference speed while maintaining high accuracy by optimizing the generation process.
- We perform comprehensive experiments on the nuScenes dataset, demonstrating that our ViLaD framework outperforms existing VLM-based methods in autonomous driving, achieving a state-of-the-art average L2 error of 1.81 m and a near-zero failure rate.
- We deploy our framework on a real-world vehicle, bridging the gap between research and practical application. In real-time driving tasks, our model demonstrated its practical value and readiness by achieving a around 95.18% success rate with an inference latency of 0.18 seconds.

II. Related Works

A. VLM-based End-to-End Autonomous Driving

VLMs are being increasingly applied to end-to-end autonomous driving systems. An effective application is using VLMs to autoregressively predict future waypoints or actions. For example, DriveGPT4 was proposed as a GPT-style model to predict encoded future scene tokens [7], and another work has focused on generating multi-step control actions guided by trajectory predictions [8].

Beyond simple generation, the reasoning capabilities of VLMs are being leveraged to build more robust and interpretable autonomous driving systems. To this end, researchers have developed question-answering frameworks where the model can explain its decisions [9]. The reasoning process has been further improved through Chain-of-Thought (CoT) prompting, where the model generates intermediate logical steps before reaching a final decision, a technique explored by DriveCoT [10]. VLM frameworks like EMMA and OpenEMMA have also demonstrated effectiveness and generalizability across challenging driving scenarios [11], [12]. Furthermore, some approaches use VLMs in a dual-process framework, combining a powerful but slower VLM for analytical reasoning with a faster, lightweight model for real-time control, mimicking human cognition to enhance performance and adaptability [13]. Other works use VLMs to provide high-level guidance or "meta-actions" to a separate end-to-end driving model, enhancing its scene understanding and ability to handle novel situations [14], [15], [16]. However, these current VLM-based autonomous driving systems heavily rely on autoregressive architectures that generate decisions through sequential, next-token prediction [17]. This paradigm introduces inherent limitations; the token-by-token generation process leads to high inference latency, and its unidirectional nature restricts the model's ability to consider future constraints, making it less suitable for dynamic, safety-critical environments [5].

B. Diffusion Language Model

Diffusion models, which have demonstrated remarkable success in visual domains by generating high-quality images [18], [19], [20], have recently emerged as a potential alternative to LLMs [6], [21]. The application of these models to natural language processing has required solving the challenge of adapting a framework designed for continuous data, like images, to the discrete nature of text [21]. Current prominent approaches include replacing the continuous diffusion process with discrete counterparts that define new forward and reverse dynamics for textual data [22].

This has led to the development of various discrete diffusion models, among which masked diffusion models (MDMs) have shown particular promise and scalability [21], [23]. MDMs operate through a process of progressively masking tokens in a sequence and then training a model to predict these masked tokens, thereby reversing the process [21]. Recent research has demonstrated the potential of scaling MDMs to sizes comparable to prominent autoregressive LLMs [21], [24]. For instance, LLaDA, an 8-billion parameter masked diffusion model, has achieved performance on par with leading LLMs such as LLaMA 3 on various benchmarks [21]. This work has shown that MDMs can be effectively scaled and can exhibit strong in-context learning and instruction-following capabilities. Furthermore, the bidirectional nature of MDMs allows them to address inherent limitations of autoregressive models, such as the "reversal curse," where models struggle with tasks that require reasoning in a non-sequential order [5], [21]. Additionally, some approaches have focused on fine-tuning existing autoregressive models using an MDM framework [25]. While still an emerging area, the progress in scaling and refining diffusion-based language models suggests they are a viable and promising direction for the future of generative AI.

III. ViLaD: Vision Language Diffusion

We present ViLaD (shown in Figure 1), a novel architecture that leverages masked diffusion models for end-to-end autonomous driving. Our architecture consists of four key components: a vision encoder for camera-based perception, a multilayer perceptron (MLP) connector for multimodal

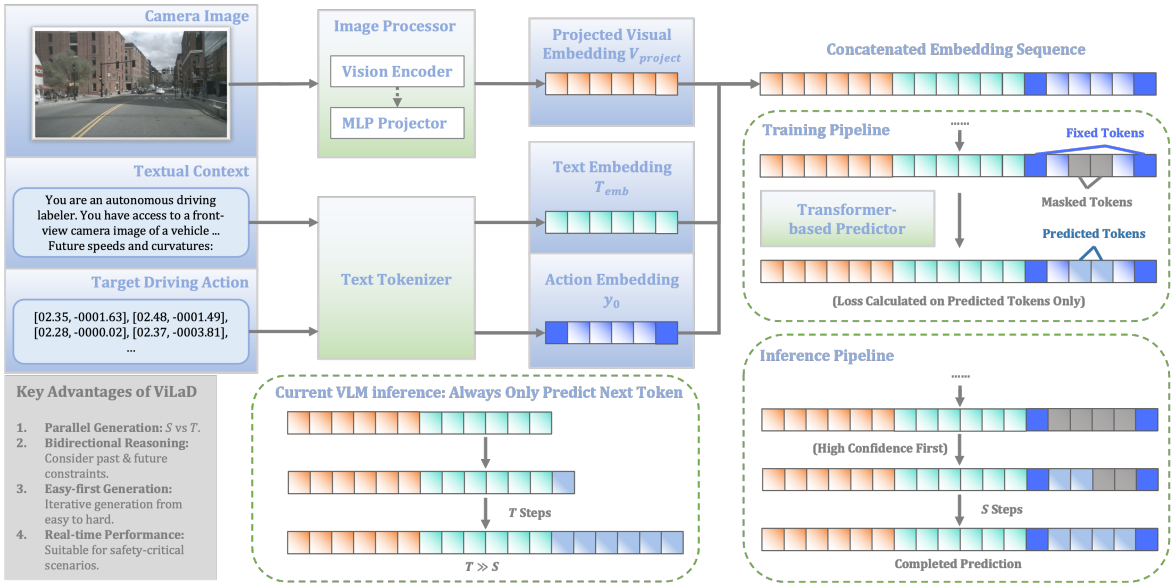

Fig. 1. This diagram shows the complete ViLaD framework for end-to-end autonomous driving: The left side illustrates multimodal input processing, where camera images and textual context are encoded and concatenated with action embeddings. The right side demonstrates the core innovation with training and inference pipelines: during training, only driving decision tokens are masked while visual/textual inputs remain unchanged; during inference, the model progressively demasks predictions starting with high-confidence decisions first. The bottom highlights ViLaD's key advantages over autoregressive VLMs, including parallel generation, bidirectional reasoning, easy-first generation, and computational efficiency.

alignment, and a diffusion-based language backbone for decision generation.

A. Motivation

1) Limitations of Autoregressive Vision Language Models: Current state-of-the-art VLMs for autonomous driving mostly employ autoregressive generation, where the probability of generating a driving action sequence $y = (y_1, y_2, ..., y_T)$ with length T given visual and text input x is factorized as:

$$P(y|x) = \prod_{i=1}^{T} P(y_i|y_1, y_2, ..., y_{i-1}, x). \tag{1}$$

This formulation introduces several fundamental limitations for autonomous driving. Specifically, the sequential dependency requires T forward passes for generating a complete driving decision. Therefore, when the length of the driving sequence is large, the inference will require a large time cost, and the inference latency will be unacceptable. For real-time autonomous driving systems requiring seconds to respond [26], this sequential bottleneck becomes prohibitive.

In addition, the next-token prediction from the autoregressive model inherently constrains the model to left-to-right information flow, preventing the integration of future trajectory constraints into current decision-making. Mathematically, this limitation can be expressed as:

$$P(y_i|y_1, \ldots, y_{i-1}, x) \neq P(y_i|y_1, \ldots, y_{i-1}, y_{i+t}, x). \tag{2}$$

2) Vision Language Diffusion Models in Autonomous Driving: Diffusion models address these fundamental limitations through parallel generation and bidirectional reasoning. The reverse process generates all tokens simultaneously according to:

$$p_\theta(y_0|y_t, x) = \prod_{i \in \{i|y_t^i = [M]\}} p_\theta(y_0^i|y_t, x). \tag{3}$$

This reduces the number of inference steps from T to S, where S represents the number of diffusion steps. Since $S \ll T$ typically, this provides a significant speedup, enabling real-time performance for autonomous driving applications.

Apart from that, the diffusion formulation enables bidirectional dependencies through $p_\theta(y_0^i|y_t, x) = f_\theta(y_t, x)$, where f_θ can access the complete generated sequence context without causal constraints. This mathematical property allows the model to consider both historical observations and future planning constraints.

A particularly compelling advantage demonstrated by our ViLaD is the progressive, easy-to-hard generation pattern enabled by the iterative denoising process. This CGEFP (Confidence-Guided Easy-First Planning) strategy allows the model to focus on high-confidence predictions first, then progressively generate more challenging aspects. Mathematically, this can be expressed as the model learning a difficulty-aware generation order $\pi(i)$ where easier tokens are predicted with higher confidence scores $p_\theta(y_o^i|y_t, x)$ at earlier denoising steps. For example, when approaching a stop sign, the model can first establish the certain decision "vehicle will stop at stop sign location (x_{stop}, y_{stop})" with high confidence, then generates the approach waypoints "vehicle follows path $(x_1, y_1) \rightarrow (x_2, y_2) \rightarrow (x_{stop}, y_{stop})$" representing the trajectory leading to the stop sign, and finally generates the most challenging decisions about post-stop navigation such as "turn left and merge into target lane" with detailed waypoint sequences. Mathematically, this can be formalized as a confidence-ordered generation sequence where $p_\theta(y_{stop}|y_{ts}, x)$ at early denoising

steps, followed by $p_\theta(y_{approach\_path}|y_{ta}, x)$ for pre-stop trajectory planning, and finally $p_\theta(y_{post\_action}|y_{tp}, x)$ for complex post-stop maneuvers, where $p_\theta(y_{stop}|y_{ts}, x) \geq p_\theta(y_{approach\_path}|y_{ta}, x) \geq p_\theta(y_{post\_action}|y_{tp}, x)$.

B. Overall Architecture Design

As shown in Figure 1, ViLaD follows an encoder-decoder architecture where visual inputs are first processed through a vision encoder, then projected into the language embedding space through an MLP connector, concatenated with textual driving context, and finally processed by a transformer-based mask predictor to predict masks. Unlike traditional autoregressive VLMs that generate tokens sequentially, ViLaD generates complete driving action sequences through parallel diffusion-based generation, enabling real-time performance while maintaining bidirectional reasoning capabilities.

The architecture takes as input multimodal driving data: camera images $I \in R^{H\times W\times 3}$ and textual driving context T (navigation instructions, traffic rules). The output consists of structured driving decisions y representing waypoints, actions, and control commands. Mathematically, our model learns the conditional distribution $P_\theta(y|I, T)$ through the diffusion framework.

C. Vision Encoder and MLP Connector

The vision encoder processes camera inputs to extract spatial-temporal features, which are then projected into the language model's embedding space for multimodal processing. The SigLP-2 [27] is used as the base vision encoder to process input images I, producing visual tokens $V \in R^{N_v\times d_{model}}$, where N_v represents the number of visual tokens and d_{model} is the hidden dimension. Then, the visual features V are projected into the language model's embedding space through a two-layer MLP connector, and the projected visual tokens $V_{project} = MLP(V)$ are concatenated with textual driving context tokens T_{emb} to form a unified multimodal input sequence:

$$X_{input} = \text{Concat}([V_{project}, T_{emb}]), \tag{4}$$

where T_{emb} represents the embedded textual driving context.

D. Diffusion Language Backbone

The core of ViLaD is a masked diffusion transformer that processes the concatenated multimodal tokens through bidirectional self-attention, enabling joint reasoning over visual and textual information.

1) Template-Constrained Action Masking in Training: The backbone employs a bidirectional transformer without causal masking, processing the concatenated multimodal sequence through transformer layers. The bidirectional self-attention enables the model to consider the complete context from both visual and textual modalities when predicting individual driving decisions.

During training, we utilize the masked diffusion approach specifically adapted for multimodal conditional generation and end-to-end autonomous driving tasks. Specifically, our input is a multimodal input X_{input} consisting of visual features V and textual driving context T. The target output y_0 is a template-constrained driving sequence that includes both control actions and punctuation marks, where punctuation marks always appear in predefined positions and the length of the y_0 is always consistent. This design ensures the model focuses solely on predicting action values rather than unnecessary formatting. We construct the model input as $X_t = \text{Concat}([X_{input}, y_0])$, applying masks only to the action tokens in y_0 while keeping X_{input} and the punctuation tokens visible throughout training.

In each training step, for a training sample $X_t = \text{Concat}([X_{input}, y_0])$, we sample a noise level $t \sim U(0, 1)$ and independently mask each token in the target action tokens with probability t, i.e., $q(y_t|y_0) = \prod_{i=1}^{L} q(y_t^i|y_0^i)$, where L is the length of the driving sequence y_0. The projected visual features $V_{project}$, textual context T, and punctuation marks remain unmasked throughout the process.

The training objective is:

$$\mathcal{L}(\theta) = -\frac{1}{t}\sum_{i=1}^{L} \mathbb{1}[y_t^i = \text{[M]}] \log p_\theta(y_0^i|X_t), \tag{5}$$

where $\mathbb{1}[y_t^i = \text{[M]}]$ ensures the loss is computed only on masked driving decision tokens, and X_t represents the full concatenated sequence with masked driving responses.

Algorithm 1 Remasking Process

Input: Current masked sequence y_t, predictions $\hat{y}_0$, total number of tokens L, total steps S, threshold τ
Output: Next sequence state y_s

```
confidence_i ← p_θ(ŷ_0^i | X_t) for i = 1, ..., L
masked_indices ← {i | y_{t_i} = [M]}
n_unmask_schedule ← ⌈L/S⌉ ▷ Number of tokens to unmask based on schedule
n_unmask_threshold ← |{i ∈ masked_indices | confidence_i > τ}|
                     ▷ Number of tokens to unmask based on confidence
n_unmask ← max(n_unmask_schedule, n_unmask_threshold)
conf_masked ← {i ∈ masked_indices | confidence_i ∈ topk(confidence, n_unmask)}
                     ▷ Get top-n confident tokens
for i ∈ conf_masked do
    y_s^i ← ŷ_0^i    ▷ Unmask the selected predictions
```

2) Confidence-Guided Easy-First Planning: During inference, we perform the reverse diffusion process starting from a partially masked sequence y_1, where all target action tokens are masked while punctuation tokens and length remain fixed, consistent with the training setup. The model then iteratively refines the masked tokens over up to S denoising steps to generate the final driving action sequence.

One primary contribution we achieve is focusing on addressing quality degradation in parallel token generation caused by the conditional independence assumption in diffusion LLMs. When unmasking multiple token positions i and j, MDMs sample these from $p_\theta(y_0^i|y_t) \cdot p_\theta(y_0^j|y_t)$ due to the conditional independence assumption, while the true joint probability is $p_\theta(y_0^i, y_0^j|y_t) = p_\theta(y_0^i|y_t) \cdot p_\theta(y_0^j|y_t, y_0^i)$. This discrepancy degrades generation quality. To address this, enlightened by Fast-dLLM [28], we propose confidence-guided easy-first planning: at each iteration, we compute confidence

TABLE I

END-TO-END PLANNING PERFORMANCE ON THE NUSCENES DATASET (* RESULTS FROM [12]).

Method	Learning	L2 (m) 1 s (↓)	L2 (m) 2 s (↓)	L2 (m) 3 s (↓)	L2 (m) Avg (↓)	Failure Rate (%) (↓)
LLaVA-1.6-7B*	Zero-Shot	1.66	3.54	4.54	3.24	4.06
Llama-3.2-11B*		1.50	3.44	4.04	3.00	23.92
Qwen2-VL-7B*		1.22	2.94	3.21	2.46	24.00
ViLaD-Zero		1.09	2.51	3.74	2.45	0.03
LLaVA-1.6-7B*	Open-EMMA	1.49	3.38	4.09	2.98	6.12
Llama-3.2-11B*		1.54	3.31	3.91	2.92	22.00
Qwen2-VL-7B*		1.45	3.21	3.76	2.81	16.11
ViLaD-CoT		1.24	2.74	4.13	2.71	0.17
LLaVA-1.6-7B	SFT	0.91	2.50	3.44	2.28	55.25
Llama-3.2-11B		0.80	2.31	3.10	2.07	0.06
Qwen2-VL-7B		1.32	2.94	3.98	2.74	0.03
ViLaD-SFT		**0.79**	**1.92**	2.83	1.85	**0.00**
ViLaD-Opt	Efficient Optimized SFT	0.81	1.93	**2.69**	**1.81**	**0.00**

scores for each token and only unmask those exceeding the threshold τ. If the number of tokens with confidence scores exceeding τ is less than $\lceil L/S \rceil$, we unmask the $\lceil L/S \rceil$ highest-confidence tokens to ensure progress.

The inference steps begin with sampling. At a step with the current sequence X_t, we use the model to predict all masked tokens simultaneously:

$$\hat{y}_0^i = \arg\max_{y_0^i} p_\theta(y_0^i \mid X_t) \quad \text{for all } i. \tag{6}$$

Then, following the threshold-based remasking strategy, we keep at least $\lceil L/S \rceil$ high-confidence predictions unchanged, and remask all the other tokens. The remask process is summarized in Algorithm 1. After remasking, the resulting sequence is used as the input sequence in the next step. This iterative process lasts for at most S steps until all tokens are masked, following the standard discretized reverse diffusion schedule. Both the number of sampling steps S and the threshold τ can be adjusted to balance between generation quality and inference speed. The whole inference process is summarized in Algorithm 2.

Algorithm 2 ViLaD Inference

Input: Visual features V, textural driving context T, driving sequence length L, steps S
Output: Predicted driving sequence y_0

Initialize: $X_1 \leftarrow \text{Concat}([\text{MLP}(\text{VE}(V)), \text{TE}(T), [\text{M}]^L])$
while y_s contains [M] **do** ▷ *Loop until all tokens are unmasked*
  $\hat{y}_0 \leftarrow \arg\max_{y_0} p_\theta(y_0|X_t)$ ▷ *Parallel prediction*
  $y_s \leftarrow \text{Remask}(y_t, \hat{y}_0, p_\theta(\hat{y}_0^i|X_t))$ ▷ *Remask Process*
  $X_t \leftarrow \text{Concat}([V_{project}, T_{emb}, y_s])$
return y_0

IV. EXPERIMENT AND RESULTS

A. Experiment Setup

We conduct experiments on the nuScenes dataset [29] using only front camera images from the last frame and ground-truth trajectories for training and validation. Evaluation focuses on L2 error in the first 3 seconds and inference latency for real-time performance. Following OpenEMMA [12], a prediction is considered a failure if the L2 distance exceeds 10 within the first second of the future trajectory. We compare ViLaD against three state-of-the-art VLMs: LLaVA-1.6-Mistral-7B [30], Llama-3.2-11B-Vision-Instruct [31], and Qwen2-VL-7B-Instruct [32] under zero-shot, OpenEMMA prompting [12], and supervised fine-tuning (SFT) settings.

We evaluate multiple configurations of our proposed ViLaD framework to analyze different aspects of performance and capabilities. We utilize a pretrained LLaDA-V [6] model as our base model, and fine-tune from it. Other ViLaD variants include ViLaD-SFT (full model with diffusion-based SFT), ViLaD-Zero (zero-shot), ViLaD-CoT (chain-of-thought reasoning using OpenEMMA [12]), and ViLaD-Opt (optimized inference), offering a broad evaluation of performance and capabilities.

B. Optimization Strategy

The outcome from dLLM-Cache [33] is used in our work. dLLM-Cache [33] introduces a training-free adaptive caching framework that combines long-interval prompt caching with partial response updates guided by feature similarity. These methods accelerate the inference speed of ViLaD and hence reduce the latency of the whole framework.

C. End-to-End Planning

Table I presents comprehensive comparison results across different methods. Our ViLaD framework demonstrates the best performance across all evaluation metrics compared to baseline VLM approaches. In the zero-shot setting, ViLaD-Zero achieves remarkable performance with L2 errors of 1.09 m, 2.51 m, and 3.74 m at 1 s, 2 s, and 3 s horizons, respectively, significantly outperforming the best baseline (Qwen2-VL-7B [32]), which achieves 1.22 m, 2.94 m, and 3.21 m. Most notably, ViLaD-Zero maintains an extremely low failure rate of 0.03% compared to baseline failure rates ranging from 4.06% to 24.00%, demonstrating the inherent robustness of our diffusion-based approach.

The OpenEMMA prompting framework shows consistent advantages of our approach. ViLaD-CoT achieves competitive performance with an average L2 error of 2.71 m while maintaining a near-zero failure rate (0.17%), outperforming all baseline methods in this category. However, we found that the ViLaD model shows no improvement using the chain-of-thought prompting method. The chain-of-thought prompting method still needed to be explored.

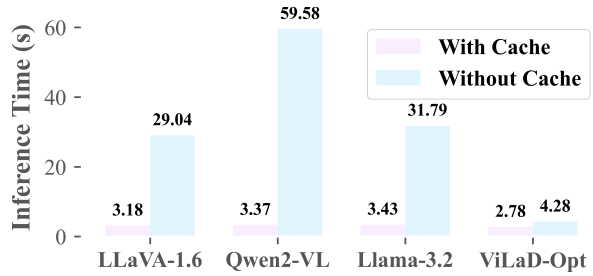

Fig. 2. The comparison of inference time between VLMs and ViLaD-Opt.

With only the SFT setting (no TCAM and CGEFP), ViLaD-SFT achieves the best overall performance with L2 errors of 0.79 m, 1.92 m, and 2.83 m across the three time horizons, resulting in an average L2 error of 1.85 m with zero failure rate. This represents significant improvements over the best baseline (Llama-3.2-11B [31] SFT), which achieves 2.07 m average L2 error. Notably, while some baseline methods achieve competitive L2 errors, they suffer from substantially higher failure rates (up to 55.25% for LLaVA-1.6-7B [30] SFT), highlighting the safety advantages of our approach.

ViLaD-Opt, our optimized deployment configuration with TCAM and CGEFP, maintains the most competitive performance (1.81 m average L2 error) while achieving inference speedup, making it suitable for real-time autonomous driving applications.

D. Inference Efficiency Experiment

To validate the real-time performance of our framework, we conducted a comprehensive inference efficiency experiment, pitting our approach against leading autoregressive VLM baselines that benefit from highly mature optimization technologies. The VLM baselines (LLaVA-1.6-7B [30], Qwen2-VL-7B [32], and Llama-3.2-11B [34]) are accelerated using standard KV caching, while our ViLaD-Opt model, representing a still-developing class of vision language diffusion models, utilizes a dLLM cache. The results from the experiment, conducted on a single NVIDIA A100 GPU and presented in Figure 2, show the significant computational advantages of our diffusion-based method. Without caching optimizations, ViLaD-Opt achieves an inference time of 4.28 s, proving to be 6.5 to 14 times faster than the autoregressive models, which require between 28.10 s and 60.18 s. While caching accelerates all models, ViLaD-Opt maintains its superiority with an inference time of 2.78 s, still outperforming the fastest baseline (3.18 s). This experiment empirically confirms that the parallel generation paradigm of ViLaD provides a critical advantage in inference speed by avoiding the token-by-token processing bottleneck in autoregressive architectures, making it highly suitable for deployment in safety-critical, real-time applications.

E. Effectiveness of Template-Constrained Action Masking

As shown in Figure 3, without TCAM optimization, increasing diffusion steps from 16 to 128 improves L2 error from 1.96 m to 1.85 m but increases inference time from 4.34 s to 25.52 s. This trade-off makes higher step counts impractical for real-time applications. With TCAM optimization, we achieve huge speedup and improved accuracy across all step configurations. For example, at 16 steps, inference time reduces from 4.34 s to 3.12 s while we achieve better L2 error (1.85 m vs 1.96 m). At 128 steps, the optimization reduces inference time from 25.52 s to 21.23 s (17% improvement) while also improving accuracy (1.78 m vs 1.85 m). The results indicate that TCAM optimization provides consistent benefits, with the most improvements at lower step counts. 32 steps with TCAM optimization provide an optimal balance between accuracy and efficiency, achieving under 2 m L2 error with inference times under 3 seconds.

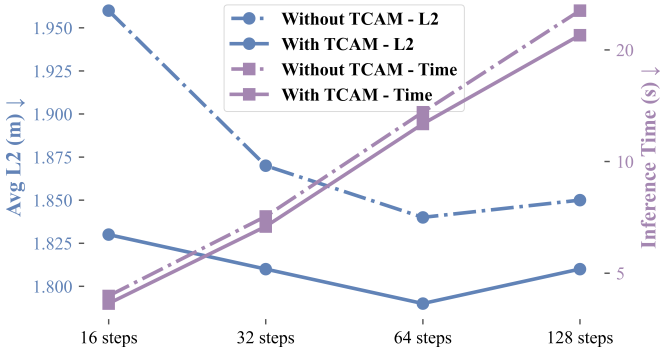

Fig. 3. The effectiveness of our Template-Constrained Action Masking (TCAM) optimization strategy across different numbers of diffusion steps.

TABLE II
THE ABLATION STUDY OF OUR CONFIDENCE THRESHOLD OPTIMIZATION STRATEGY ACROSS DIFFERENT THRESHOLDS.

Confidence Threshold	Avg L2 (m) (↓)	Inference Time (s) (↓)
0.9	1.82	3.95
0.7	1.87	3.29
0.5	**1.81**	2.78
0.3	1.97	**2.73**

F. Ablation Study on Confidence Threshold in CGEFP

We conduct detailed ablation studies to understand the impact of key hyperparameters on system performance. The confidence threshold analysis shows important insights about the trade-off between accuracy and efficiency in our CGEFP strategy. As shown in Table II, a threshold of 0.5 achieves the best balance with 1.81 m average L2 error and 2.78 s inference time. Higher thresholds (0.9, 0.7) tend to be more conservative, unmasking fewer tokens per iteration, which can lead to slightly higher accuracy (0.9: 1.82 m; 0.7: 1.87 m) but at the cost of increased inference time (0.9: 3.95 s; 0.7: 3.29 s). Lower thresholds (0.3) unmask more tokens aggressively, reducing inference time to 2.73 s but potentially degrading accuracy to 1.97 m due to premature commitment to low-confidence predictions.

V. Real-World Case Studies: On-Board Interactive Parking

To further validate the practical applicability of our approach, we present two complementary real-world case

TABLE III

ON-BOARD MODEL PERFORMANCE METRICS FOR INTERACTIVE PARKING.

Scenario	Model Performance		Safety Metrics		Comfort Metrics	
	Latency (s)	Success Rate (%)	Long. Vel. Var.(m^2/s^2)	Lat. Vel. Var.(10^{-4} m^2/s^2)	Max Long. Accel.(m/s^2)	Max Lat. Accel.(m/s^2)
Parking Selection	0.18	95.18	1.07	2.66	1.43	1.12
E2E Instruction Following	0.32	85.91	0.62	1.53	1.39	0.68

studies: interactive parking and instruction-following autonomous driving. Both are deployed on the same embedded vehicle platform, but are evaluated on different driving tasks.

A. Vehicle and System Setup

The experiment is performed on a production vehicle equipped with a custom sensor suite and computational hardware, as illustrated in the vehicle setup in Figure 4. The on-board processing is handled by a Spectra ECU featuring an Intel i9-9900 CPU and an NVIDIA Quadro RTX-A4000 GPU. Inputs for the task are captured by in-cabin cameras with microphones, which receive the driver's commands. The system leverages a pre-recorded map of the parking area and integrates with the vehicle's existing autonomous driving stack for final motion planning and control.

To evaluate system performance, we measure both efficiency and driving quality. Inference latency is used to assess the feasibility of real-time interaction. The success rate quantifies how reliably the model interprets and carries out instructions. A trial is considered successful if the whole driving decision aligns with the instruction; otherwise, it is labeled unsuccessful. Safety is evaluated through the variance of longitudinal and lateral velocities. Passenger comfort is captured by recording maximum longitudinal and lateral accelerations during each maneuver.

We fine-tune two specialized models based on the pre-trained models from SMDM [24], which are lightweight and support more on-board inference compared with the larger-scale models used in the full framework above. Each model is used for one of the real-world tasks. For the interactive parking case, we collect a dataset of approximately 3,000 command–action pairs describing diverse parking slot preferences. For the instruction-following case, we construct a dataset of a similar scale, focusing on short-horizon driving commands. For evaluation, we collect a separate test set of 300 unseen instructions for each task, allowing us to assess generalization under realistic command diversity.

B. Case Study I: Interactive Parking

Our first case focuses on selecting a parking slot based on natural language commands (e.g., "Find a spot near the exit"). The target driving action predicted by our method includes the final waypoint corresponding to the chosen slot. As reported in Table III, ViLaD achieves an average inference latency of 0.18 s and a 95.18% success rate, while ensuring stable dynamics (longitudinal velocity variance: $1.07\ \mathrm{m^2\,s^{-2}}$, lateral variance: $2.66\times10^{-4}\ \mathrm{m^2\,s^{-2}}$, suggesting that the vehicle maintains smooth motion profiles without oscillations or abrupt shifts during the maneuver. Comfort is maintained with maximum longitudinal and lateral accelerations of $1.43\ \mathrm{m\,s^{-2}}$ and $1.12\ \mathrm{m\,s^{-2}}$, respectively, values that

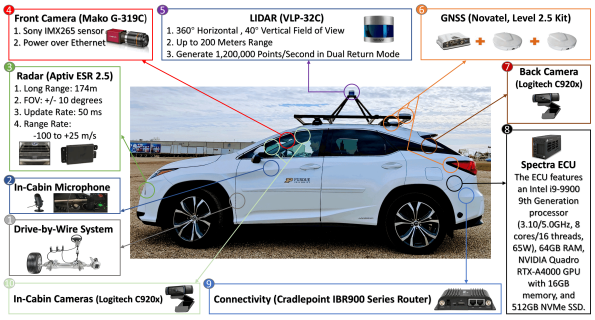

Fig. 4. Overview of the autonomous vehicle setup.

fall within thresholds commonly regarded as imperceptible to passengers [35], [36]. These results highlight the system's ability to perform fine-grained spatial reasoning and safe vehicle maneuvering in constrained environments.

C. Case Study II: End-to-End Instruction Following

The second case evaluates our method's ability to follow human-issued driving commands in free-road conditions. At the start of each trial, the driver specifies a natural language instruction such as "Turn left and stop near the crosswalk" or "Continue straight and yield at the intersection." The vehicle is then required to autonomously execute the command for the next 28.5 m to 37.6 m of driving.

The results (also in Table III) indicate a success rate of 85.91% with an average inference latency of 0.32 s, confirming real-time feasibility. Safety analysis shows that the longitudinal velocity variance remains as low as 0.62 $\mathrm{m^2\,s^{-2}}$, while lateral variance is 1.53×10^{-4} $\mathrm{m^2\,s^{-2}}$, reflecting stable path tracking and reliable control even under more dynamic, instruction-driven conditions. For the comfort, the maximum accelerations are limited to $1.39\ \mathrm{m\,s^{-2}}$ in the longitudinal direction and $0.68\ \mathrm{m\,s^{-2}}$ laterally. These values are significantly below the commonly cited discomfort thresholds [35], [36]. Overall, these outcomes demonstrate that ViLaD generalizes beyond parking scenarios to more dynamic instruction-following tasks, while sustaining safe and comfortable driving behavior.

VI. CONCLUSION

We present ViLaD, a novel end-to-end autonomous driving framework that replaces conventional autoregressive models with a Large Vision Language Diffusion approach. By leveraging masked diffusion, ViLaD enables parallel action generation, includes bidirectional context, and adopts an easy-first decision strategy to reduce latency and improve robustness. Experiments on the nuScenes benchmark demonstrate that ViLaD outperforms state-of-the-art vision-language models in both accuracy and efficiency, achieving near-zero failure rates in dynamic scenarios. The successful

on-board deployment for a real-world interactive parking task further proves the practical value and real-time performance of our approach, underscoring its potential for real-world automotive applications.

REFERENCES

[1] C. Cui, Y. Ma, X. Cao, W. Ye, Y. Zhou, K. Liang, J. Chen, J. Lu, Z. Yang, K.-D. Liao, *et al.*, "A survey on multimodal large language models for autonomous driving," *arXiv preprint arXiv:2311.12320*, 2023.

[2] X. Tian, J. Gu, B. Li, Y. Liu, C. Hu, Y. Wang, K. Zhan, P. Jia, X. Lang, and H. Zhao, "DriveVLM: The Convergence of Autonomous Driving and Large Vision-Language Models," *arXiv*, 2024.

[3] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K. K. Y. Wong, Z. Li, and H. Zhao, "DriveGPT4: Interpretable End-to-end Autonomous Driving via Large Language Model," Oct. 2023, arXiv:2310.01412.

[4] C. Cui, Y. Ma, X. Cao, W. Ye, and Z. Wang, "Drive as You Speak: Enabling Human-Like Interaction with Large Language Models in Autonomous Vehicles," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*. Waikoloa, HI, USA: IEEE, Jan. 2024, pp. 902–909.

[5] L. Berglund, M. Tong, M. Kaufmann, M. Balesni, A. C. Stickland, T. Korbak, and O. Evans, "The Reversal Curse: LLMs trained on" A is B" fail to learn" B is A"," *arXiv preprint arXiv:2309.12288*, 2023.

[6] Z. You, S. Nie, X. Zhang, J. Hu, J. Zhou, Z. Lu, J.-R. Wen, and C. Li, "Llada-v: Large language diffusion models with visual instruction tuning," *arXiv preprint arXiv:2505.16933*, 2025.

[7] 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.

[8] P. Wu, X. Jia, L. Chen, J. Yan, H. Li, and Y. Qiao, "Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6119–6132, 2022.

[9] 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, pp. 8186–8193, Oct. 2024, conference Name: IEEE Robotics and Automation Letters.

[10] T. Wang, E. Xie, R. Chu, Z. Li, and P. Luo, "DriveCoT: Integrating Chain-of-Thought Reasoning with End-to-End Driving," Mar. 2024, arXiv:2403.16996. [Online]. Available: http://arxiv.org/abs/2403.16996

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

[12] 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 *Proceedings of the Winter Conference on Applications of Computer Vision*, 2025, pp. 1001–1009.

[13] J. Mei, Y. Ma, X. Yang, L. Wen, X. Cai, X. Li, D. Fu, B. Zhang, P. Cai, M. Dou, B. Shi, L. He, Y. Liu, and Y. Qiao, "Continuously Learning, Adapting, and Improving: A Dual-Process Approach to Autonomous Driving," Oct. 2024, arXiv:2405.15324 [cs]. [Online]. Available: http://arxiv.org/abs/2405.15324

[14] B. Jiang, S. Chen, B. Liao, X. Zhang, W. Yin, Q. Zhang, C. Huang, W. Liu, and X. Wang, "Senna: Bridging large vision-language models and end-to-end autonomous driving," *arXiv preprint arXiv:2410.22313*, 2024.

[15] Z. Guo, K. Gubernatorov, S. Asfaw, Z. Yagudin, and D. Tsetserukou, "Vdt-auto: End-to-end autonomous driving with vlm-guided diffusion transformers," *arXiv preprint arXiv:2502.20108*, 2025.

[16] H. Fu, D. Zhang, Z. Zhao, J. Cui, D. Liang, C. Zhang, D. Zhang, H. Xie, B. Wang, and X. Bai, "Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation," *arXiv preprint arXiv:2503.19755*, 2025.

[17] L. Chen, Z. Wang, S. Ren, L. Li, H. Zhao, Y. Li, Z. Cai, H. Guo, L. Zhang, Y. Xiong, *et al.*, "Next token prediction towards multimodal intelligence: A comprehensive survey," *arXiv preprint arXiv:2412.18619*, 2024.

[18] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. pmlr, 2015, pp. 2256–2265.

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

[20] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456*, 2020.

[21] 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," *arXiv preprint arXiv:2502.09992*, 2025.

[22] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. Van Den Berg, "Structured denoising diffusion models in discrete state-spaces," *Advances in neural information processing systems*, vol. 34, pp. 17 981–17 993, 2021.

[23] A. Lou, C. Meng, and S. Ermon, "Discrete diffusion modeling by estimating the ratios of the data distribution," *arXiv preprint arXiv:2310.16834*, 2023.

[24] S. Nie, F. Zhu, C. Du, T. Pang, Q. Liu, G. Zeng, M. Lin, and C. Li, "Scaling up masked diffusion models on text," 2025. [Online]. Available: https://arxiv.org/abs/2410.18514

[25] S. Gong, S. Agarwal, Y. Zhang, J. Ye, L. Zheng, M. Li, C. An, P. Zhao, W. Bi, J. Han, *et al.*, "Scaling diffusion language models via adaptation from autoregressive models," *arXiv preprint arXiv:2410.17891*, 2024.

[26] C. Cui, Z. Yang, Y. Zhou, J. Peng, S.-Y. Park, C. Zhang, Y. Ma, X. Cao, W. Ye, Y. Feng, J. Panchal, L. Li, Y. Chen, and Z. Wang, "On-board vision-language models for personalized autonomous vehicle motion control: System design and real-world validation," 2024. [Online]. Available: https://arxiv.org/abs/2411.11913

[27] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, O. Hénaff, J. Harmsen, A. Steiner, and X. Zhai, "Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features," 2025. [Online]. Available: https://arxiv.org/abs/2502.14786

[28] C. Wu, H. Zhang, S. Xue, Z. Liu, S. Diao, L. Zhu, P. Luo, S. Han, and E. Xie, "Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding," 2025. [Online]. Available: https://arxiv.org/abs/2505.22618

[29] 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, pp. 11 621–11 631.

[30] C. Li, C. Wong, S. Zhang, N. Usuyama, H. Liu, J. Yang, T. Naumann, H. Poon, and J. Gao, "LLaVA-Med: Training a Large Language-and-Vision Assistant for Biomedicine in One Day," June 2023, arXiv:2306.00890 [cs]. [Online]. Available: http://arxiv.org/abs/2306.00890

[31] H. Zhang, X. Li, and L. Bing, "Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding," in *EMNLP*. arXiv, 2023, arXiv:2306.02858 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2306.02858

[32] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin, "Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution," 2024. [Online]. Available: https://arxiv.org/abs/2409.12191

[33] Z. Liu, Y. Yang, Y. Zhang, J. Chen, C. Zou, Q. Wei, S. Wang, and L. Zhang, "dllm-cache: Accelerating diffusion large language models with adaptive caching," 2025. [Online]. Available: https://arxiv.org/abs/2506.06295

[34] H. Zhang, X. Li, and L. Bing, "Video-llama: An instruction-tuned audio-visual language model for video understanding," 2023. [Online]. Available: https://arxiv.org/abs/2306.02858

[35] K. N. de Winkel, T. Irmak, R. Happee, and B. Shyrokau, "Standards for passenger comfort in automated vehicles: Acceleration and jerk," *Applied Ergonomics*, vol. 106, p. 103881, 2023.

[36] J. Xu, K. Yang, Y. Shao, and G. Lu, "An experimental study on lateral acceleration of cars in different environments in sichuan, southwest china," *Discrete Dynamics in nature and Society*, vol. 2015, no. 1, p. 494130, 2015.