

WAM-Diff2: Hierarchical AR-to-Diffusion Distillation for Highly Efficient Autonomous Driving VLA

Zhihao Zhu^{1*}, Hanlin Shang^{1*}, Mingwang Xu^{1*}, Feipeng Cai^{2*}, Zhuolin He¹,
Yaoyi Li², Jianhua Han², Hang Xu², Siyu Zhu^{1†}

¹Fudan University, ²Yingwang Intelligent Technology Co.,Ltd

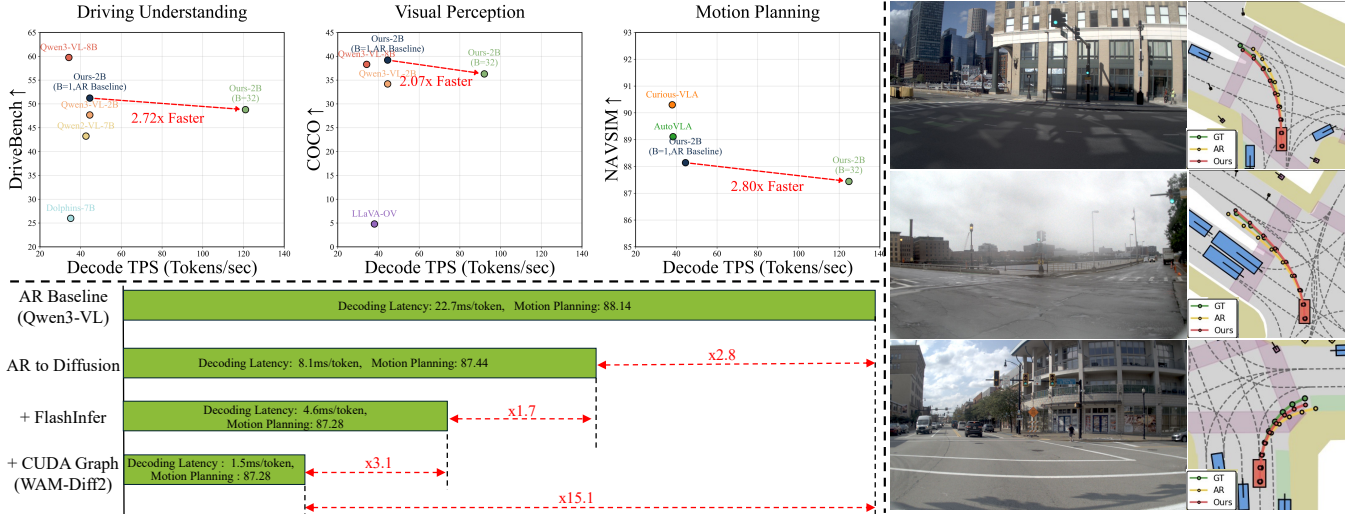


Figure 1: Efficiency, multi-task performance, and planning robustness of WAM-Diff2. **Top left:** performance versus decoding latency (ms/token) across driving understanding (DriveBench), visual perception (COCO), and motion planning (NAVSIM). Our parallel discrete diffusion approach ($B = 32$) yields up to a $2.80\times$ speedup over the autoregressive baseline while preserving high multi-task proficiency. **Bottom left:** latency optimization breakdown. Transitioning from the AR baseline to the WAM-Diff2 diffusion paradigm accelerates decoding by $2.8\times$. Integrating FlashInfer and CUDA Graphs yields a cumulative $15.1\times$ latency reduction (from 22.7 ms/token to 1.5 ms/token) with negligible planning degradation. **Right:** mitigation of exposure bias in long-horizon trajectory prediction. Compared to the AR baseline, which suffers from compounding drift errors, WAM-Diff2 delivers highly robust, temporally consistent planning that closely aligns with the ground truth.

Abstract

Vision-Language-Action (VLA) models have emerged as a prominent paradigm for end-to-end autonomous driving; however, their efficient deployment is severely constrained by high computational latency and exposure bias arising from sequential autoregressive decoding. Conversely, while specialized diffusion policies enable low-latency, parallel execution, training them from scratch typically yields narrow, single-task architectures that lack holistic visual-linguistic reasoning. Successfully transforming pre-trained autoregressive generalists into parallel diffusion models could combine multi-task cognitive intelligence with execution efficiency, yet this transition presents a formidable architectural challenge due to mismatched attention patterns (causal versus bidirectional) and divergent optimization objectives. To bridge this divide,

we introduce WAM-Diff2, a multi-task discrete diffusion VLA framework powered by a three-stage hierarchical distillation strategy. By structuring the architectural shift through progressive block-wise adaptation, block-wise distillation, and model-wise cross-scale distillation, WAM-Diff2 preserves the underlying semantic foundations of the base model while accelerating inference. Extensive evaluations across driving understanding, perception, and planning benchmarks demonstrate that WAM-Diff2 effectively mitigates exposure bias and achieves performance parity with autoregressive baselines. Crucially, the autoregressive-to-diffusion transition yields a $2.8\times$ decoding speedup, which scales to an ultimate $15.1\times$ acceleration when combined with system-level optimizations including FlashInfer and CUDA Graphs.

1 Introduction

Vision-Language-Action models (Sima et al. 2024; Hwang et al. 2024; Li et al. 2025b; Chen, Wang, and Zhang 2025; Li et al. 2025c; Han et al. 2025; Li et al. 2025a; Cui et al.

*Equal contribution

†Corresponding author

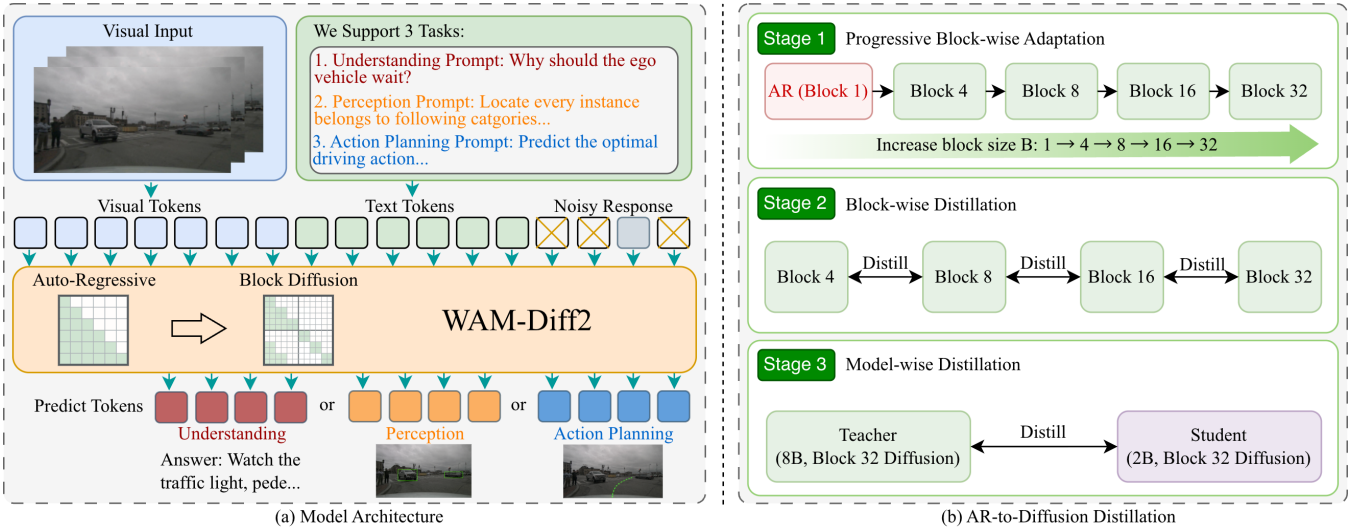


Figure 2: Architectural overview of WAM-Diff2 and the three-stage hierarchical distillation framework. (a) Multi-view images, linguistic instructions, and noisy response tokens are mapped into a unified token space for multi-task inference across driving understanding, perception, and action planning. By replacing traditional autoregressive causal masks with a block-causal attention pattern, WAM-Diff2 enables bidirectional parallel token refinement within each decoding block while preserving cross-block causal constraints. (b) The autoregressive-to-diffusion transition is accomplished through three progressive stages: Stage 1 incrementally expands the decoding block size ($B = 1 \rightarrow 32$) to smoothly adapt the underlying attention mechanisms; Stage 2 applies recursive block-wise distillation from a stable, small-block diffusion teacher to guide larger-block student variants; and Stage 3 employs model-wise cross-scale distillation to transfer holistic semantic capabilities from an advanced 8B diffusion teacher to a highly efficient 2B student.

2025; Zhou et al. 2025a,b; Li et al. 2026b) have emerged as the leading paradigm for unified end-to-end autonomous driving, consolidating diverse tasks—ranging from linguistic scene understanding (Li et al. 2024a; Bai et al. 2025a; Han et al. 2025; Li et al. 2025c) to 2D/3D perception (Han et al. 2025; Li et al. 2026b; Bai et al. 2025a) and motion planning (Wu et al. 2022; Hu et al. 2023; Jiang et al. 2023; Chitta et al. 2022; Li et al. 2025c; Wang et al. 2025b; Li et al. 2025b; Zou et al. 2025; Kim et al. 2026)—within a single network architecture. Despite their impressive multi-task proficiency, prevailing driving generalists are predominantly built upon causal autoregressive foundations. This design choice imposes fundamental limitations on their operational deployment. Structurally, the sequential nature of autoregressive decoding generates outputs token-by-token, incurring high computational latency that threatens real-time driving safety. Algorithmically, autoregressive training relies on ground-truth prefixes via teacher forcing (Cen et al. 2024; Gu et al. 2023; Agarwal et al. 2024; Bachmann and Nagarajan 2024), leaving the model inherently vulnerable to compounding errors during long-horizon inference—a phenomenon widely recognized as exposure bias.

To overcome these latency and robustness bottlenecks, the community has bifurcated into two disconnected lines of research. On one side, autoregressive generalists (Hwang et al. 2024; Li et al. 2025c; Chen, Wang, and Zhang 2025; Han et al. 2025; Zhou et al. 2025b) leverage large language model scales to achieve broad multi-task semantic reasoning at the expense of computational efficiency and causal fragility. On the opposite side, planning-centric diffusion models (Zheng et al. 2025; Liao et al. 2025; Zou et al. 2025;

Xu et al. 2025a) utilize highly parallel, discrete or continuous diffusion processes to achieve low-latency control and robust bidirectional trajectory refinement. However, training these specialized architectures from scratch typically yields narrow, single-task systems that lack the holistic understanding and visual-linguistic reasoning capabilities of their autoregressive counterparts. Consequently, existing paradigms present a rigid compromise between multi-task cognitive intelligence and parallelized execution efficiency.

In this work, we transcend this dichotomy by introducing a universal architecture translation paradigm rather than presenting a localized, planning-only diffusion model. We argue that the vast semantic knowledge and multi-task reasoning embedded within mature, pre-trained autoregressive models (Li et al. 2024a; Bai et al. 2025a; Zhu et al. 2025) should be leveraged, rather than discarded, to initialize parallel generation architectures. The core challenge lies in the radical paradigm gap between sequential, unidirectional causal pre-training and parallel, bidirectional context refinement. This transition presents a formidable architectural challenge due to mismatched attention patterns and divergent optimization objectives.

To bridge this fundamental divide, we present WAM-Diff2, a multi-task discrete diffusion VLA framework powered by a novel three-stage hierarchical distillation strategy. Instead of initiating training from scratch, our framework seamlessly transforms any advanced, pre-trained autoregressive driving generalist into an ultra-efficient parallel refinement architecture. Our core insight is that the architectural transition must be staged to protect the underlying semantic foundations. In Stage I (Progressive Block-Wise Adaptation), we design a

block-causal attention mechanism that incrementally relaxes rigid causal constraints through a curriculum-based expansion of decoding blocks, ensuring mathematical stability during the attention shift. In Stage II (Block-Wise Distillation), we actively expose the student to intermediate noisy states generated during parallel decoding, utilizing an on-policy (Agarwal et al. 2024; Gu et al. 2023), small-block diffusion teacher to enforce paradigm consistency and eliminate exposure bias. In Stage III (Model-Wise Cross-Scale Distillation), we address the capacity-efficiency trade-off by scaling knowledge transfer from an advanced 8B diffusion teacher to a compact 2B student. Crucially, we discover that models sharing the same diffusion paradigm exhibit highly aligned token prediction patterns, allowing the smaller model to easily inherit complex reasoning from the larger teacher without compromising parallel decoding speed.

Comprehensive evaluations across trajectory planning (NAVSIM (Dauner et al. 2024), Bench2Drive (Jia et al. 2024)), driving-oriented visual question answering (DriveBench (Xie et al. 2025), LingoQA (Marcu et al. 2024)), and visual perception (COCO (Lin et al. 2014)) benchmarks demonstrate that WAM-Diff2 achieves strict performance parity or superiority relative to its autoregressive foundation while delivering an obvious inference speedup. Specifically, the autoregressive-to-diffusion transition yields a $2.8\times$ decoding speedup, which scales to an ultimate $15.1\times$ acceleration when combined with system-level optimizations including FlashInfer and CUDA Graphs. Ultimately, our framework establishes a highly scalable, low-cost pipeline capable of transforming existing and future state-of-the-art autoregressive VLAs into high-throughput, highly efficient deployable diffusion agents.

2 Related Works

VLA for Autonomous Driving. Vision-Language-Action models have evolved from foundational scene understanding and VQA (Xu et al. 2024; Tian et al. 2024; Sima et al. 2024; Ma et al. 2024; Nie et al. 2024; Wang et al. 2025a) toward unified end-to-end control (Zhou et al. 2025a,b; Fu et al. 2025; Mao et al. 2023; Shao et al. 2024; Jiang et al. 2024; Xu et al. 2025b). Current driving generalists, such as EMMA (Hwang et al. 2024), Percept-WAM (Han et al. 2025), DriveMoE (Yang et al. 2025b), and UniDriveVLA (Li et al. 2026b), integrate perception and planning within autoregressive frameworks (Mao et al. 2023; Chen, Wang, and Zhang 2025; Zhou et al. 2025a,b; Li et al. 2026b). However, these sequential architectures are inherently limited by unidirectional causal constraints, high inference latency, and exposure bias.

Discrete Diffusion Models. Discrete diffusion provides a robust alternative for sequential generation (Austin et al. 2021; Hoogetboom et al. 2021; Campbell et al. 2022; Li et al. 2022; Gong et al. 2022; Lou, Meng, and Ermon 2023; Shi et al. 2024; Sahoo et al. 2024; Gat et al. 2024). Recently, masked diffusion has scaled to large-scale language and multimodal domains (Nie et al. 2025; Ye et al. 2025a; You et al. 2025; Yang et al. 2025a; Ou et al. 2024; Arriola et al. 2025; Von Rütte et al. 2025), enabling parallel iterative refinement. While applied to trajectory generation in robotics and driving (Cui et al. 2025; Li et al. 2025a; Xu et al. 2025a; Dang et al. 2026;

Zhang et al. 2026; Liang et al. 2025; Wen et al. 2025b; Ye et al. 2025a; Wen et al. 2025a; Liu et al. 2026; Chen et al. 2026), existing diffusion models often lack the multi-task breadth of autoregressive counterparts. We bridge this capability gap by transforming a unified autoregressive VLA into a discrete diffusion framework.

LLM and LVM Distillation. Knowledge distillation (Hinton, Vinyals, and Dean 2015) is critical for model compression. Traditional off-policy distillation for autoregressive models (Kim and Rush 2016; Sanh et al. 2019) suffers from exposure bias (Bengio et al. 2015; Ranzato et al. 2015), which recent on-policy approaches (Agarwal et al. 2024; Li et al. 2026c; Afsharrad et al. 2026) and divergence-based objectives (Gu et al. 2023) seek to mitigate. Transferring knowledge from autoregressive teachers to diffusion students is uniquely challenging due to the discrepancy between causal and bidirectional attention. Our tiered framework addresses this through progressive adaptation and cross-scale alignment.

3 Methodology

Overview. We propose a VLA architecture that seamlessly transitions from standard autoregressive sequence generation to highly parallel discrete diffusion. This section first formalizes the problem by contrasting the autoregressive foundation with our discrete diffusion formulation (Section 3.1). We then introduce our core contribution: a three-stage distillation framework designed to transfer the multi-task autoregressive VLA teacher to a highly efficient, parallel diffusion student (Section 3.2). Finally, we detail the network architecture and training protocols (Section 3.3).

3.1 Preliminaries

VLA Formulation. Our baseline model Φ is a multi-task VLA generalist that maps multimodal sensory streams and linguistic instructions into a unified discrete output space. Given a set of N_c surrounding multi-camera RGB images $\mathcal{I}^{\text{cam}} = \{I_k\}_{k=1}^{N_c}$ and a natural-language task instruction $\mathcal{L}^{\text{instruct}}$, the network generates a task-conditioned sequence: $\mathcal{O} = \Phi(\mathcal{I}^{\text{cam}}, \mathcal{L}^{\text{instruct}})$. We represent all multi-task targets using a shared, unified text tokenizer, deliberately omitting task-specific projection heads. This framework encapsulates three core autonomous driving capabilities: (1) *Driving Understanding*: Given a linguistic query $\mathcal{L}^{\text{qa}}$, the agent aligns visual features with text semantics to produce a natural-language scene-reasoning response $\mathcal{O}^{\text{qa}}$. (2) *Visual Perception*: For spatial grounding tasks, the model outputs a serialized sequence of M bounding boxes $\mathcal{O}^{\text{det}} = \{(b_i, c_i, s_i)\}_{i=1}^M$, where $b_i \in \mathbb{R}^4$ denotes bounding box coordinates, c_i is the class label, and s_i represents the confidence score. (3) *Motion Planning*: Conditioned on a navigation command $\mathcal{L}^{\text{nav}}$ and the ego-vehicle status $\mathcal{L}^{\text{ego}}$, the network generates a future trajectory $\mathcal{O}^{\text{traj}} = \{(\hat{x}_t, \hat{y}_t)\}_{t=1}^T$ comprising T BEV waypoints.

Autoregressive VLA. Conventional driving generalists serialize these multi-task outputs into a flat sequence of discrete tokens $\mathcal{O} = (o_1, \dots, o_L)$. The network optimizes the parameters by maximizing the joint likelihood via a standard causal

language modeling objective:

$$p(\mathcal{O} \mid \mathcal{I}^{\text{cam}}, \mathcal{L}^{\text{instruct}}) = \prod_{i=1}^L p(o_i \mid o_{<i}, \mathcal{I}^{\text{cam}}, \mathcal{L}^{\text{instruct}}) \quad (1)$$

where $o_{<i}$ represents the causal prefix history. While this causal formulation leverages the scaling laws of pre-trained vision-language models, its sequential token-by-token generation incurs a computational complexity of $\mathcal{O}(L)$. This linear dependency presents two fundamental limitations: it introduces severe latency bottlenecks that impede real-time deployment, and its reliance on ground-truth prefixes during training (teacher forcing (Song et al. 2020)) induces exposure bias, leaving the system susceptible to compounding drift errors during closed-loop execution.

Discrete Diffusion VLA. To bypass these computational and architectural bottlenecks, we reformulate sequence generation as an iterative denoising process over a finite discrete state space $\mathcal{V}$. Given a clean target sequence $\mathbf{o}^0 = (o_1, \dots, o_L) \in \mathcal{V}^L$, we employ an absorbing-state (masked) discrete diffusion formulation defined by a forward and reverse process.

Forward Process: The forward process progressively corrupts the clean token sequence $\mathbf{o}^0$ into a noisy state $\mathbf{o}^t$ over T discrete timesteps. At each transition step, a token either retains its identity or transitions to a special absorbing [MASK] token. Representing tokens as one-hot vectors, the transition matrix Q_t is defined as:

$$q(\mathbf{o}^t \mid \mathbf{o}^{t-1}) = \prod_{i=1}^L \mathbf{o}_i^{t-1} Q_t, \quad Q_t = (1 - \beta_t)\mathbf{I} + \beta_t \mathbf{1} e_m^\top, \quad (2)$$

where e_m is the one-hot base vector for the [MASK] token, $\mathbf{I}$ is the identity matrix, and $\beta_t \in [0, 1]$ defines the noise schedule. This formulation allows the intermediate marginal distribution to be evaluated in closed form as $q(\mathbf{o}^t \mid \mathbf{o}^0) = \mathbf{o}^0 \bar{Q}_t$, where $\bar{Q}_t = \prod_{s=1}^t Q_s$, facilitating highly efficient parallel training via random state sampling.

Reverse Parallel Refinement: The denoising network Φ_θ is trained to reconstruct the clean sequence directly from the corrupted observation, modeling the conditional distribution $p_\theta(\mathbf{o}^0 \mid \mathbf{o}^t, c)$. During inference, we execute a parallel remasking strategy: starting from a fully masked sequence, the model predicts all token distributions simultaneously. At each denoising step, it fixes a subset of tokens with the highest prediction confidence and re-masks the remaining tokens, compressing full-sequence decoding complexity to $\mathcal{O}(T)$ steps, where $T \ll L$.

3.2 AR-to-Diffusion VLA Transformation

Transferring semantic knowledge from a causal autoregressive teacher to a bidirectional diffusion student introduces a fundamental paradigm gap, which arises from mismatched attention patterns (causal vs. bidirectional) and divergent optimization objectives (next-token prediction vs. iterative denoising). To bridge this gap without performance degradation, we propose a three-stage hierarchical distillation framework.

Stage I: Progressive Block-wise Adaptation. We formulate the architectural transition from sequential decoding to parallel generation as a curriculum learning problem. We define the autoregressive baseline as a limiting case of block-wise generation where the decoding block size $B = 1$. To smoothly transition to global bidirectional attention, we define a curriculum that incrementally expands the decoding block size $B \in \{4, 8, 16, 32\}$.

For a target block size B , the sequence $\mathbf{y}$ is partitioned into non-overlapping blocks. The student network p_{θ_B} is optimized to reconstruct a fully masked block $\mathcal{B} \subset \mathbf{y}$ conditioned on the visual context x , the unmasked context $\mathbf{y}_{\setminus \mathcal{B}}$, and the discrete diffusion timestep t :

$$\mathcal{L}_{\text{SFT}}^B = -\mathbb{E}_{t, \mathcal{B}: |\mathcal{B}|=B} \sum_{i \in \mathcal{B}} \log p_{\theta_B}(y_i \mid x, \mathbf{y}_{\setminus \mathcal{B}}, t). \quad (3)$$

To prevent semantic degradation during this attention shift, we employ a progressive bootstrapping initialization scheme:

$$\theta_B^{(0)} \leftarrow \theta_{B/2}^{\text{SFT}}, \quad \text{where } \theta_1 = \theta_{\text{AR}}. \quad (4)$$

By sequentially expanding B , the network progressively internalizes joint token distributions and stabilizes the transition from causal to block-causal attention mechanisms.

Stage II: Block-wise Distillation. Although Stage I structurally adapts the attention layers to larger block dimensions, optimizing exclusively via ground-truth targets provides poor guidance for iterative refinement trajectory. To instill robust parallel denoising mechanics, Stage II introduces block-wise distillation from a stable, small-block diffusion teacher. We employ progressive block-wise distillation following the sequence $4 \rightarrow 8 \rightarrow 16 \rightarrow 32$. Specifically, the block-4 model p_{θ_4} obtained in Stage I is used to distill the block-8 model p_{θ_8} . The resulting block-8 model subsequently serves as the teacher for the block-16 model, and the same procedure is repeated for the block-32 model. At each stage, the student p_{θ_B} is trained to match the predictive distribution of the preceding teacher $p_{\theta_{B/2}}$ over identical corrupted states. To prevent the mode-covering behavior associated with standard forward KL divergence—which often compromises multi-modal trajectory distributions—we align the output distributions across the target block B using a symmetric Jensen-Shannon Divergence (JSD) loss (Englsson and Azizpour 2021):

$$\mathcal{L}_{\text{BWD}}^B = \mathbb{E}_{t, \mathcal{B}, \tilde{\mathbf{y}}_t^B} \sum_{i \in \mathcal{B}} \text{JSD} \left(p_{\theta_{B/2}}^{(i)}(\cdot \mid x, \tilde{\mathbf{y}}_t^B, t) \parallel p_{\theta_B}^{(i)}(\cdot \mid x, \tilde{\mathbf{y}}_t^B, t) \right), \quad (5)$$

where $\tilde{\mathbf{y}}_t^B$ represents the noisy token sequence, and the JSD between two distributions p and q is defined as:

$$\text{JSD}(p \parallel q) = \frac{1}{2} \text{KL} \left(p \parallel \frac{p+q}{2} \right) + \frac{1}{2} \text{KL} \left(q \parallel \frac{p+q}{2} \right). \quad (6)$$

Distilling from an intermediate diffusion anchor rather than the raw autoregressive baseline encourages that the optimization signal remains strictly paradigm-consistent, isolating the learning objective to parallel token scaling rather than cross-paradigm reconciliation.

Table 1: Multi-task performance and decoding efficiency across driving generalist paradigms. We evaluate a single unified model checkpoint across three core autonomous driving capabilities: driving understanding (DriveBench, LingoQA), visual perception (COCO mAP), and motion planning (NAVSIM v1/v2 PDMS/EPDMS). Efficiency is reported as decoding throughput (Tokens Per Second, TPS) under identical hardware configurations, where the dual values for our diffusion models denote standard implementation versus hardware-optimized execution (FlashInfer + CUDA Graphs). The $\dagger$ and $*$ symbols denote external methods utilizing reinforcement learning and score-based candidate selection, respectively.

Method	Driving Understanding		Visual Perception	Motion Planning		Efficiency
	DriveBench $\uparrow$	LingoQA $\uparrow$	COCO $\uparrow$	NAVSIM V1 $\uparrow$	NAVSIM V2 $\uparrow$	Decode TPS $\uparrow$
Multi-Task VLMs and VLAs						
Qwen3-VL-2B (Bai et al. 2025a)	47.68	48.00	34.20	–	–	44.5
Qwen3-VL-8B (Bai et al. 2025a)	59.75	60.20	38.30	–	–	34.2
Qwen2-VL-7B (Wang et al. 2024)	43.25	52.60	19.30	–	–	42.6
Qwen2.5-VL-3B (Bai et al. 2025b)	30.52	52.40	18.70	–	–	37.6
Qwen2.5-VL-7B (Bai et al. 2025b)	35.55	60.00	20.10	–	–	40.2
InternVL3-2B (Zhu et al. 2025)	33.30	49.40	7.50	–	–	45.9
InternVL3-8B (Zhu et al. 2025)	37.99	45.00	17.50	–	–	42.5
LLaVA-OV (Li et al. 2024a)	21.99	38.80	4.80	–	–	38.0
Percept-WAM (Han et al. 2025)	–	–	51.70	80.40 / 90.20*	–	–
Recogdrive-8B (Li et al. 2025c)	<u>56.71</u>	67.20	–	86.50 / 89.60 $\dagger$	–	–
Ours-2B ($B = 1$, AR Baseline)	51.23	68.40	<u>39.20</u>	88.14 / 91.50*	88.32	44.5
Ours-2B ($B = 4$)	48.92	<u>68.20</u>	37.80	<u>87.87 / 91.47*</u>	<u>87.99</u>	68.3 / 401.4
Ours-2B ($B = 8$)	48.81	68.00	37.30	87.57 / 91.19*	87.78	89.5 / 561.6
Ours-2B ($B = 16$)	48.77	66.00	36.50	87.32 / 91.26*	87.30	<u>108.4 / 635.3</u>
Ours-2B ($B = 32$)	48.80	65.80	36.30	87.44 / 91.05*	87.50	124.8 / 673.4

Stage III: Model-Wise Cross-Scale Distillation. To recover the multi-task high-level intelligence lost during the compression to a 2B student, we scale the model capacity using a large-scale (8B) teacher. We observe that knowledge transfer is most effective when the teacher and student share consistent “thinking patterns”, which we quantify using the top- K overlap ratio ρ_K :

$$\rho_K(p_T, p_S) = \frac{|\text{TopK}(p_T) \cap \text{TopK}(p_S)|}{K} \quad (7)$$

Our empirical analysis shows that ρ_K is significantly higher between two diffusion models than between an autoregressive teacher and a diffusion student. Therefore, we utilize an 8B block-32 diffusion model $p_{\phi_{32}}^{8B}$ as our ultimate teacher. The 2B student $p_{\theta_B}^{2B}$ ($B = 32$) is trained to match this 8B teacher on its own trajectories:

$$\mathcal{L}_{\text{MWD}} = \mathbb{E}_{t, \mathcal{B}, \tilde{\mathbf{y}}_t^{2B}} \sum_{i \in \mathcal{B}} \text{JSD} \left(p_{\phi_{32}}^{8B, (i)}(\cdot | x, \tilde{\mathbf{y}}_t^{2B}, t) \parallel p_{\theta_B}^{2B, (i)}(\cdot | x, \tilde{\mathbf{y}}_t^{2B}, t) \right) \quad (8)$$

This stage, unlike the previous two stages, is not mandatory. It enables the compact diffusion model to inherit the complex perception, understanding, and planning capabilities of the 8B foundation while maintaining efficient parallel decoding.

3.3 Model Architecture and Training

Model Architecture. WAM-Diff2 builds upon the Qwen3-VL framework (Bai et al. 2025a), comprising a visual encoder, a cross-modal adapter, and a decoder-only Transformer backbone. We instantiate our framework at two scales: (1) *8B*

Variant: Integrates a SigLIP2-SO-400M visual encoder (27 blocks, 4096 hidden dimensions) (Tschannen et al. 2025) with 36 Transformer blocks. (2) *2B Variant*: Utilizes a SigLIP2-Large encoder (24 blocks, 2048 hidden dimensions) paired with 28 Transformer blocks. To accommodate arbitrary camera aspect ratios, we apply a dynamic resolution strategy using 2D RoPE (Su et al. 2024) to interpolate spatial coordinates. All modalities—including linguistic tokens, 2D bounding boxes, and future waypoints—are processed through a unified text tokenizer, entirely removing task-specific projection heads. Crucially, the standard causal attention mask is modified into a block-causal pattern: tokens within the same decoding block employ bidirectional parallel attention, while inter-block dependencies retain causal constraints.

System-Level Inference Optimizations. To fully exploit the parallel decoding capability of the discrete diffusion formulation, we implement two hardware-level optimizations. *FlashInfer* (Ye et al. 2025b): We deploy customized attention kernels optimized for the block-causal attention pattern, accelerating intra-block bidirectional token refinement by maximizing shared memory throughput. *CUDA Graphs* (Ghosh et al. 2025): Given that parallel remasking operates over a static number of denoising steps T with fixed tensor shapes, we encapsulate the entire execution graph. This eliminates CPU launch overheads, boosting decoding throughput.

Staged Training. The transition from an autoregressive generalist to a parallel diffusion agent proceeds through a unified multi-task pipeline spanning two key datasets (DriveLM (Sima et al. 2024), LingoQA (Marcu et al. 2024) for

Table 2: Motion planning performance on NAVSIM v1 and v2 benchmarks. [†]: Methods using reinforcement learning or reinforcement fine-tuning; *: Results obtained via score-based candidate selection (Yao et al. 2026). NAVSIM v2 results utilize the original evaluation protocol for consistency.

Method	NAVSIM v1						NAVSIM v2									
	NC [†]	DAC [†]	TTC [†]	Comf. [†]	EP [†]	PDMS [†]	NC [†]	DAC [†]	DDC [†]	TLC [†]	EP [†]	TTC [†]	LK [†]	HC [†]	EC [†]	EPDMS [†]
TransFuser (Chitta et al. 2022)	97.7	92.8	92.8	100.0	79.2	84.0	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	76.7
Hydra-MDP++ (Li et al. 2024b)	97.6	96.0	93.1	100.0	80.4	86.6	97.2	97.5	99.4	99.6	83.1	96.5	94.4	98.2	70.9	81.4
ARTEMIS (Feng et al. 2025)	98.3	95.1	94.3	100.0	81.4	87.0	98.3	95.1	98.6	99.8	81.5	97.4	96.5	98.3	–	83.1
DriveSuprim (Yao et al. 2026)	97.8	97.3	93.6	100.0	86.7	89.9	97.5	96.5	99.4	99.6	88.4	96.6	95.5	98.3	77.0	83.1
DiffusionDrive (Liao et al. 2025)	98.2	96.2	94.7	100.0	82.2	88.1	98.2	95.9	99.4	99.8	87.5	97.3	96.8	98.3	87.7	84.5
ReCogDrive [†] (Li et al. 2025c)	97.9	97.3	94.9	100.0	87.3	90.8	98.3	95.2	99.5	99.8	87.1	97.5	96.6	98.3	86.5	83.6
SGDrive-2B (Li et al. 2026a)	98.6	95.1	95.4	100.0	81.2	87.4	98.6	94.3	99.5	99.9	86.0	97.9	96.1	98.3	85.9	86.2
DriveVLA-W0 (Li et al. 2025b)	98.7	99.1	95.3	99.3	83.3	90.2	99.0	98.4	99.3	99.9	87.0	98.1	93.2	97.9	58.9	86.5
ResAD (Zheng et al. 2026)	98.0	97.5	94.1	100.0	83.3	88.8	97.8	97.2	99.5	99.8	88.2	96.9	97.0	98.4	88.2	85.5
DriveFine [†] (Dang et al. 2026)	98.6	97.9	95.2	99.9	85.5	90.7	98.7	97.3	98.8	99.8	88.2	97.8	97.7	98.4	84.7	87.1
DiffusionDriveV2 [†] (Zou et al. 2025)	98.3	97.9	94.8	99.9	87.5	91.2	97.7	96.6	99.2	99.8	88.9	97.2	96.0	97.8	91.0	85.5
Ours ($B = 1$ AR Baseline)	98.4	96.2	94.8	99.9	82.4	88.1	98.4	96.2	99.5	99.8	87.5	97.6	97.5	98.2	86.1	88.3
Ours ($B = 32$)	98.6	96.1	95.5	99.8	82.1	88.3	98.6	96.1	99.4	99.9	87.0	97.9	97.6	98.2	87.7	88.6
Ours ($B = 32$)*	99.1	98.0	96.7	99.9	85.1	<u>91.1</u>	99.1	98.0	99.4	99.8	88.0	98.5	97.2	98.2	86.4	90.7

scene reasoning; COCO (Lin et al. 2014) for object detection; NAVSIM (Dauner et al. 2024) and Bench2Drive (Jia et al. 2024) for motion planning).

Phase 1: Multi-Task Autoregressive Pretraining (Optional). The base models are trained for 5 epochs using standard causal next-token prediction to establish a robust multi-task semantic foundation across the consolidated training mixture. If a pre-trained autoregressive VLA generalist is available, this phase can be entirely bypassed, allowing the framework to directly initialize from the existing model weights.

Phase 2: Progressive Adaptation & Block-Wise Distillation. During this phase, we gradually relax the causal constraints by expanding the block size $B \in \{4, 8, 16, 32\}$, fine-tuning for five epochs. Each variant is initialized from the weights of its predecessor ($\theta_B^{(0)} \leftarrow \theta_{B/2}$). Concurrently, the larger-block $B_{student}$ students are distilled against the stable $B = B_{student}/2$ diffusion teacher using a symmetric JSD loss over intermediate noisy states, directly mitigating exposure bias. Specifically, we train the larger-block student variants for five epochs per adaptation step.

Phase 3: Model-Wise Cross-Scale Distillation. To close the capacity gap, the 2B student ($B = 32$) is distilled for 5 epochs against the advanced 8B diffusion teacher ($B = 32$). By minimizing the JSD between their predictive token distributions, the compact student effectively inherits the complex cognitive reasoning of the larger foundation model without sacrificing parallel inference throughput.

4 Experiments

4.1 Experimental Setups

Implementation Details. The architecture is built upon the Qwen3-VL (Bai et al. 2025a) foundation, with multi-view camera inputs resized to 1920×1080 pixels. All optimization phases utilize the AdamW optimizer with a weight decay of 0.05, a cosine learning rate scheduler, and a 10% linear warmup under a global batch size of 128 across 32 ASCEND

910C NPUs. Autoregressive pretraining runs for 5 epochs with a base learning rate of 4×10^{-5} . During the block-causal adaptation and block-wise distillation stages, the backbone learning rate is maintained at 4×10^{-5} while the visual encoder uses a decoupled rate of 2×10^{-6} to stabilize cross-modal alignment. The final model-wise cross-scale distillation applies a unified learning rate of 2×10^{-6} to both components.

Evaluation Benchmarks. We evaluate WAM-Diff2 across five benchmarks covering motion planning, scene understanding, and visual perception. Open-loop and closed-loop trajectory planning are assessed via NAVSIM (Dauner et al. 2024) (v1/v2) and Bench2Drive (Jia et al. 2024), reporting the standard and extended Predictive Driver Model Score (PDMS and EPDMS), Success Rate (SR), and Driving Score (DS). Multimodal driving scene reasoning is quantified on LingoQA (Marcu et al. 2024) and DriveBench (Xie et al. 2025) using Lingo-Judge and GPT Score, respectively. General spatial grounding performance is measured via mean Average Precision (mAP) on COCO (Lin et al. 2014).

Evaluation Protocols. To ensure rigorous comparison against both multi-task foundations and specialized planners, we establish two evaluation paradigms. The *unified multi-task protocol* evaluates a single frozen checkpoint simultaneously across all perception, understanding, and planning tasks to verify cross-task capability preservation under the autoregressive-to-diffusion transition. Conversely, the *task-specific protocol* trains and evaluates the architecture exclusively on isolated navigation benchmarks, ensuring fair comparisons against specialized, planning-only baselines.

4.2 Comparison with Existing Works

We evaluate WAM-Diff2 under two evaluation paradigms: a *unified multi-task protocol* (Table 1) that assesses multi-capability preservation and inference throughput using a single model, and a *task-specific protocol* (Tables 2 and 3) for

Table 3: Closed-loop evaluation and multi-ability assessment on the Bench2Drive benchmark. We report the standard driving metrics (Efficiency, Comfort, Success Rate, and Driving Score) alongside task-specific success rates (%) across safety-critical conditions to evaluate robustness against distribution shifts. **Bold** and underline denote the best and second-best results.

Method	Closed-loop Metric $\uparrow$				Multi-Ability Test (%) $\uparrow$					
	Eff.	Comf.	Succ.	DS	Merge	Overtake	Emerg. Brake	Give Way	Traf. Sign	Mean
TCP* (Wu et al. 2022)	54.26	47.80	15.00	40.70	16.18	20.00	20.00	10.00	6.99	14.63
TCP-ctrl* (Wu et al. 2022)	55.97	51.51	7.27	30.47	10.29	4.44	10.00	10.00	6.45	8.23
TCP-traj* (Wu et al. 2022)	76.54	18.08	30.00	59.90	8.89	24.29	51.67	40.00	46.28	34.22
ThinkTwice (Jia et al. 2023b)	76.93	16.22	3.13	62.44	27.38	18.42	35.82	50.00	54.23	37.17
DriveAdapter* (Jia et al. 2023a)	70.22	16.01	33.08	64.22	28.82	26.38	48.76	50.00	56.43	42.08
AD-MLP (Zhai et al. 2023)	48.45	22.63	0.00	18.05	0.00	0.00	0.00	0.00	4.35	0.87
UniAD-T. (Hu et al. 2023)	123.92	47.04	13.18	40.73	8.89	9.33	20.00	20.00	15.43	14.73
UniAD-B. (Hu et al. 2023)	129.21	43.58	16.36	45.81	14.10	17.78	21.67	10.00	14.21	15.55
VAD (Jiang et al. 2023)	157.94	46.01	15.00	42.35	8.11	24.44	18.64	20.00	19.15	18.07
ReCogDrive (Li et al. 2025c)	138.18	17.45	45.45	71.36	29.73	20.00	<u>69.09</u>	20.00	71.34	42.03
Orion (Fu et al. 2025)	151.48	17.38	54.62	77.74	25.00	<u>71.11</u>	78.33	33.00	69.15	54.72
DriveMoE (Yang et al. 2026)	<u>175.96</u>	15.31	48.64	74.22	34.67	40.00	65.45	40.00	59.44	47.91
UniDriveVLA (Li et al. 2026b)	198.86	11.78	51.82	78.37	38.75	80.00	50.00	30.00	58.95	51.53
Ours-2B (B=1, AR Baseline)	118.10	21.40	51.96	80.51	45.59	62.79	60.34	50.00	89.25	61.59
Ours-2B (B=4)	116.24	22.01	51.36	<u>80.01</u>	47.50	60.00	60.00	50.00	<u>88.42</u>	<u>61.18</u>
Ours-2B (B=8)	118.17	19.26	50.23	<u>79.73</u>	<u>46.25</u>	62.22	60.00	50.00	87.37	61.17
Ours-2B (B=16)	110.45	21.36	50.45	79.52	45.00	60.00	61.67	50.00	87.37	60.81
Ours-2B (B=32)	108.20	23.15	49.55	78.93	43.75	62.22	61.67	50.00	85.26	60.58

equitable comparison against specialized driving planners.

Unified Multi-Task Performance. As shown in Table 1, our 2B diffusion model with a block size of $B = 32$ achieves a DriveBench score of 48.80 and a LingoQA metric of 65.80, matching or exceeding the semantic capabilities of mature autoregressive generalists sharing the same backbone. For spatial perception, the model retains strong visual perception with 36.3 mAP on COCO, showing that dense cross-modal alignment remains intact through the generative paradigm transition. Furthermore, without relying on auxiliary reinforcement fine-tuning or score-based candidate selection, WAM-Diff2 delivers a robust open-loop planning performance of 87.44 PDMS on NAVSIM v1. These results demonstrate that transitioning to parallel discrete diffusion successfully maintains holistic scene reasoning and planning precision while unlocking substantial gains in decoding efficiency.

Task-Specific Motion Planning. Under the planning isolated evaluation framework, WAM-Diff2 preserves the planning capability of its autoregressive counterpart, demonstrating that the conversion from autoregressive to diffusion decoding introduces no meaningful performance degradation. On the open-loop NAVSIM benchmark (Table 2), our model matches or surpasses the original autoregressive baseline, achieving 88.3 PDMS (v1) and 88.6 EPDMS (v2). When coupled with score-based candidate selection (*), WAM-Diff2 reaches a peak performance of 91.1 PDMS and 90.7 EPDMS, outperforming advanced specialized driving agents such as ReCogDrive (Li et al. 2025c) and DriveVLA-W0 (Li et al. 2025b). In closed-loop evaluations on Bench2Drive (Table 3), WAM-Diff2 demonstrates robust generalization under extreme distribution shifts, attaining a Success Rate of 49.55%

and a Driving Score of 78.93. Crucially, it yields pronounced improvements in safety-critical scenarios like Emergency Braking (61.67%) and Traffic Sign compliance (85.26%), underscoring the benefits of our distillation framework in mitigating inference-time exposure bias. Corresponding trajectory rollouts are visualized in Figures 5 and 6.

Block-Size Scaling and Efficiency. Table 1 further quantifies the latency-accuracy Pareto frontier across varied decoding block sizes B . Expanding B from 4 to 32 yields substantial hardware parallelization: vanilla implementation throughput rises from 68.3 to 124.8 TPS, which scales dramatically to a peak of 673.4 TPS when integrated with system-level FlashInfer and CUDA Graph optimizations. This represents a cumulative $15.1\times$ inference acceleration over the sequential autoregressive baseline. Importantly, this throughput optimization introduces negligible performance degradation: motion-planning metrics remain remarkably stable ($\Delta \leq 0.5$ points on NAVSIM v1/v2), while open-ended visual question answering and grounding exhibit only minor trade-offs.

4.3 Ablation and Discussion

Effectiveness of AR-to-Diffusion Transformation Stages. We quantify the contribution of each stage in our framework using the NAVSIM benchmark (Dauner et al. 2024) (Table 4). Specifically, direct AR-to-diffusion adaptation results in a significant performance drop (84.1 PDMS) due to the radical shift from causal to bidirectional attention. Block-wise distillation recovers substantial performance (87.7) by aligning the student with a stable, small-block diffusion teacher. Model-wise distillation further closes the gap, allowing the 2B student to inherit high-level reasoning from the 8B teacher, ultimately achieving 88.3 PDMS.

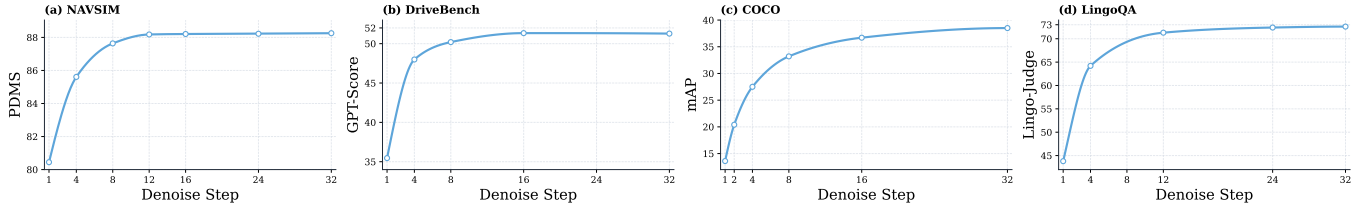


Figure 3: Effect of denoising steps. PDMS improves as the number of denoising steps increases and gradually saturates at approximately 8–16 steps.

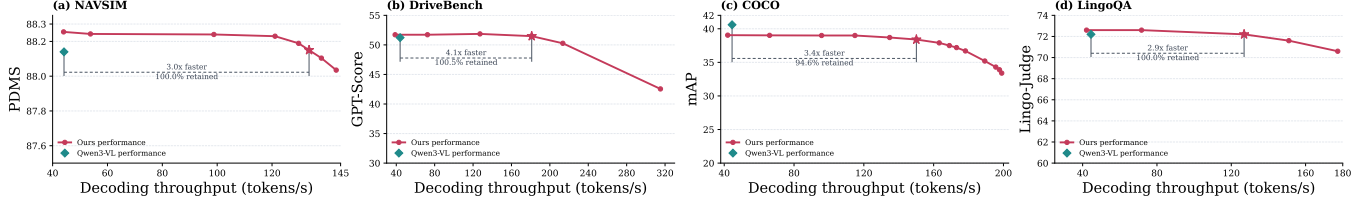


Figure 4: Performance–efficiency trade-offs of WAM-Diff2 across multiple benchmarks. WAM-Diff2 consistently preserves driving understanding, general visual perception, and motion-planning performance while achieving higher decoding throughput.

Table 4: Cumulative ablation of AR-to-diffusion VLA distillation. Starting from the AR baseline, we compare direct AR-to-diffusion adaptation, block-wise distillation, and model-wise distillation. All diffusion variants are evaluated under the same decoding configuration on NAVSIM.

Setting	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
AR baseline	98.4	96.2	94.8	99.9	82.4	88.1
+ Direct AR-to-diffusion adaptation	97.9	93.0	93.2	99.3	78.5	84.1
+ Block-wise distillation	98.2	95.7	95.0	99.3	81.6	87.7
+ Model-wise distillation	98.6	96.1	95.5	99.8	82.1	88.3

Table 6: Top- K overlap ratio between different teachers and the diffusion student on training samples. All values are reported as percentages. Higher ρ_K indicates stronger teacher–student alignment.

Teacher	$\rho_1 \uparrow$	$\rho_5 \uparrow$	$\rho_{10} \uparrow$	$\rho_{20} \uparrow$
AR Teacher	83.6	51.2	54.3	46.1
Diffusion Teacher	84.8	58.2	60.5	50.7



Figure 5: Qualitative results of WAM-Diff2 on NAVSIM. The predicted trajectories demonstrate robust planning behavior across diverse driving scenarios.

Paradigm-Consistent Distillation Mechanics. To optimize knowledge transfer across generative paradigms, we evaluate both the teacher configuration and the token-level alignment objectives. As shown in Table 6, the 8B diffusion teacher exhibits significantly higher top- K token overlap ($\rho_5 = 58.2\%$) with the 2B diffusion student compared to the 8B autoregressive teacher ($\rho_5 = 51.2\%$). This structural alignment translates to a severe performance drop when

Table 5: Ablation of divergence objectives for AR-to-diffusion VLA distillation. Specifically, we compare forward KL, reverse KL, and Jensen–Shannon divergence (JSD) for motion planning on the NAVSIM benchmark, while keeping all other training and evaluation settings fixed.

Distillation Loss	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
Forward KL	98.6	96.0	95.6	99.9	81.9	88.2
Reverse KL	98.5	95.9	95.3	99.7	81.8	88.0
JSD	98.6	96.1	95.5	99.8	82.1	88.3

Table 7: Ablation of different teacher models for motion planning on the NAVSIM benchmark. We compare autoregressive teacher and diffusion teacher under otherwise identical distillation settings.

Teacher Model	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
AR Teacher	97.2	89.5	91.7	97.3	74.6	79.9
Diffusion Teacher	98.6	96.1	95.5	99.8	82.1	88.3



Figure 6: Qualitative results of WAM-Diff2 on Bench2Drive. The predicted trajectories demonstrate robust planning behavior across diverse driving scenarios.

distilling from the autoregressive teacher (79.9 PDMS; Table 7), showing that paradigm consistency is prerequisite for effective cross-scale distillation. Regarding the alignment objective (Table 5), while forward and reverse KL divergences yield comparable baselines, the symmetric JSD achieves peak performance (88.3 PDMS) by preserving both the mode-seeking precision and broad semantic coverage required for multi-modal trajectory distributions.

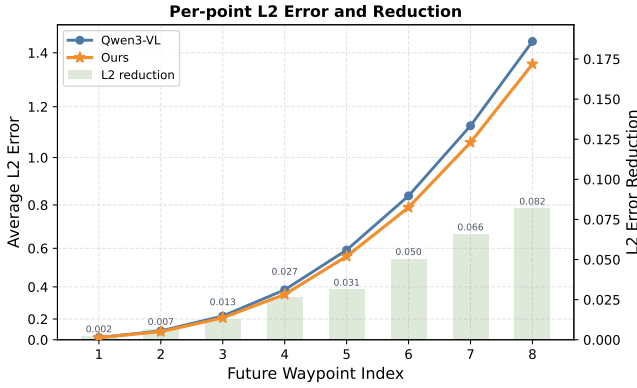


Figure 7: Per-waypoint L_2 error under inference on the NAVSIM dataset. The L_2 errors of the autoregressive baseline gradually increase compared to our method, and there is a trend of error amplification. These results suggest that bidirectional diffusion generation help to alleviate error accumulation caused by autoregressive exposure bias.

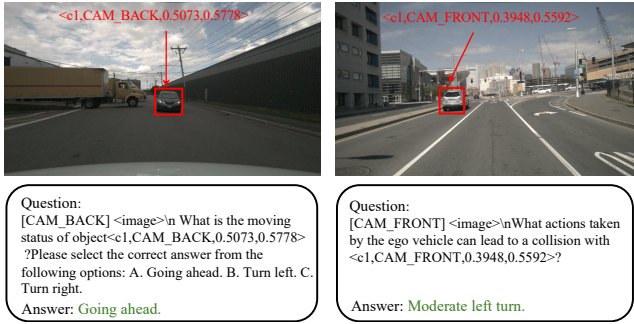


Figure 8: Visualization of understanding results.



Figure 9: Visualization of perception results.

Speed–Accuracy Pareto Frontier. We analyze the inference dynamics of WAM-Diff2 across varying refinement budgets and decoding block sizes (B). As illustrated in Figure 3, multi-task performance scales rapidly with additional denoising steps, saturating at approximately 8–16 iterations across all benchmarks. Figure 4 further maps the performance–efficiency trade-off: trajectory planning and driving understanding remain robust across a wide throughput range, whereas visual grounding (COCO mAP) exhibits earlier degradation due to its strict coordinate-formatting constraints. By adjusting B from 4 to 32 (Table 1), vanilla

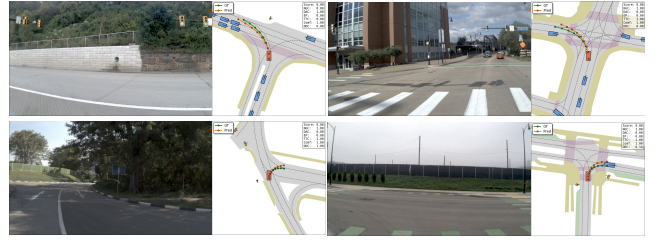


Figure 10: Failure cases visualization on NAVSIM.

decoding throughput rises from 68.3 to 124.8 TPS with negligible planning degradation ($\Delta \leq 0.5$ PDMS), demonstrating flexible, runtime-configurable control over execution latency.

Efficiency Analysis. The throughput acceleration of WAM-Diff2 stems from unlocking parallel block-wise token refinement. Under a standard implementation, replacing sequential AR decoding with block-wise diffusion provides an immediate $2.8\times$ speedup on NAVSIM (Figure 1). Integrating hardware-level optimizations yields compound gains: specialized Flash-Infer attention kernels maximize shared memory throughput to provide a $1.7\times$ acceleration, while CUDA Graphs eliminate CPU launch overheads for static tensor shapes, contributing an additional $3.1\times$ boost. Combined, these algorithmic and system optimizations deliver a cumulative $15.1\times$ reduction in decoding latency (from 22.7 ms/token to 1.5 ms/token).

Mitigation of Exposure Bias. WAM-Diff2 leverages bidirectional attention and iterative refinement to denoise tokens within each block, allowing the model to dynamically revise uncertain intermediate predictions rather than permanently committing to early mistakes. To quantify this architectural advantage, we evaluated WAM-Diff2 against the autoregressive baseline across 12,146 paired NAVSIM samples. As shown in Figure 7, the block-wise diffusion framework reduces the average per-waypoint L_2 error by 5.8% (from 0.5935 to 0.5589). Crucially, the absolute error reduction widens monotonically over time—from 0.002 at the initial waypoint to 0.082 at the final horizon (waypoint 8)—demonstrating that bidirectional parallel refinement effectively mitigates long-horizon error accumulation and enhances closed-loop robustness against compounding sequence drift.

Limitations and Failure Cases. Qualitative failure modes on the NAVSIM (Dauner et al. 2024) and Bench2Drive (Jia et al. 2024) benchmarks are visualized in Figures 10 and 11. Despite its high inference throughput and robustness against exposure bias, WAM-Diff2 exhibits two primary limitations. First, its structural reliance on discrete tokenization can introduce spatial quantization artifacts, occasionally compromising the smoothness required for high-precision trajectory planning. Second, the downstream capabilities of the parallel diffusion student remain fundamentally upper-bounded by the baseline semantic reasoning proficiency of the initial autoregressive teacher.

5 Conclusion

We presented WAM-Diff2, a universal architecture translation paradigm that seamlessly converts pre-trained, multi-task



Figure 11: Failure cases visualization on Bench2Drive.

autoregressive VLAs into high-throughput discrete diffusion agents for autonomous driving. By deploying a novel three-stage hierarchical distillation strategy—spanning progressive block-wise adaptation, block-wise distillation, and cross-scale paradigm alignment—our framework successfully bridges the architectural chasm between sequential causal generation and parallel bidirectional refinement. Extensive evaluations across motion planning, visual perception, and scene understanding benchmarks demonstrate that WAM-Diff2 preserves the holistic cognitive reasoning of large-scale autoregressive generalists while eliminating their structural bottlenecks. Algorithmically, the transition to a block-causal diffusion mechanism successfully mitigates exposure bias, monotonically suppressing long-horizon error accumulation during sequence rollouts. Under system-level optimizations via FlashInfer and CUDA Graphs, WAM-Diff2 compresses token decoding latency from 22.7 ms to 1.5 ms, yielding a cumulative $15.1\times$ inference acceleration. Ultimately, this work provides a scalable, low-cost pipeline to deploy advanced multimodal driving intelligence under the rigid throughput constraints of highly efficient autonomous systems.

References

- Afsharrad, A.; Abedsoltan, A.; Moradipari, A.; and Lall, S. 2026. On-Policy Distillation of Language Models for Autonomous Vehicle Motion Planning. *arXiv preprint arXiv:2604.07944*.
- Agarwal, R.; Vieillard, N.; Zhou, Y.; Stanczyk, P.; Garea, S. R.; Geist, M.; and Bachem, O. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*.
- Arriola, M.; Gokaslan, A.; Chiu, J. T.; Yang, Z.; Qi, Z.; Han, J.; Sahoo, S. S.; and Kuleshov, V. 2025. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*.
- Austin, J.; Johnson, D. D.; Ho, J.; Tarlow, D.; and Van Den Berg, R. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34: 17981–17993.
- Bachmann, G.; and Nagarajan, V. 2024. The pitfalls of next-token prediction. *arXiv preprint arXiv:2403.06963*.
- Bai, S.; Cai, Y.; Chen, R.; Chen, K.; Chen, X.; Cheng, Z.; Deng, L.; Ding, W.; Gao, C.; Ge, C.; et al. 2025a. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.
- Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; Zhong, H.; Zhu, Y.; Yang, M.; Li, Z.; Wan, J.; Wang, P.; Ding, W.; Fu, Z.; Xu, Y.; Ye, J.; Zhang, X.; Xie, T.; Cheng, Z.; Zhang, H.; Yang, Z.; Xu, H.; and Lin, J. 2025b. Qwen2.5-VL Technical Report. *arXiv:2502.13923*.
- Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28.
- Campbell, A.; Benton, J.; De Bortoli, V.; Rainforth, T.; Deligiannidis, G.; and Doucet, A. 2022. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35: 28266–28279.
- Cen, Z.; Liu, Y.; Zeng, S.; Chaudhari, P.; Rangwala, H.; Karypis, G.; and Fakoor, R. 2024. Bridging the training-inference gap in llms by leveraging self-generated tokens. *arXiv preprint arXiv:2410.14655*.
- Chen, J.; Song, W.; Chen, S.; Wang, J.; Li, Z.; and Li, H. 2026. DFM-VLA: Iterative Action Refinement for Robot Manipulation via Discrete Flow Matching. *arXiv preprint arXiv:2603.26320*.
- Chen, Y.; Wang, Y.; and Zhang, Z. 2025. Drivinggpt: Unifying driving world modeling and planning with multi-modal autoregressive transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 26890–26900.
- Chitta, K.; Prakash, A.; Jaeger, B.; Yu, Z.; Renz, K.; and Geiger, A. 2022. TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving. *arXiv:2205.15997*.
- Cui, C.; Zhou, Y.; Peng, J.; Park, S.-Y.; Yang, Z.; Sankaranarayanan, P.; Zhang, J.; Zhang, R.; and Wang, Z. 2025. Vilad: A large vision language diffusion framework for end-to-end autonomous driving. *arXiv preprint arXiv:2508.12603*.
- Dang, C.; Ang, S.; Li, Y.; Tian, H.; Wang, J.; Li, G.; Ye, H.; Ma, J.; Chen, L.; and Wang, Y. 2026. DriveFine: Refining-Augmented Masked Diffusion VLA for Precise and Robust Driving. *arXiv preprint arXiv:2602.14577*.
- Dauner, D.; Hallgarten, M.; Li, T.; Weng, X.; Huang, Z.; Yang, Z.; Li, H.; Gilitschenski, I.; Ivanovic, B.; Pavone, M.; et al. 2024. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37: 28706–28719.
- Engleson, E.; and Azizpour, H. 2021. Generalized jensen-shannon divergence loss for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34: 30284–30297.
- Feng, R.; Xi, N.; Chu, D.; Wang, R.; Deng, Z.; Wang, A.; Lu, L.; Wang, J.; and Huang, Y. 2025. Artemis: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving. *IEEE Robotics and Automation Letters*, 11(1): 226–233.
- Fu, H.; Zhang, D.; Zhao, Z.; Cui, J.; Liang, D.; Zhang, C.; Zhang, D.; Xie, H.; Wang, B.; and Bai, X. 2025. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. *arXiv preprint arXiv:2503.19755*.

- Gat, I.; Remez, T.; Shaul, N.; Kreuk, F.; Chen, R. T.; Synnaeve, G.; Adi, Y.; and Lipman, Y. 2024. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385.
- Ghosh, A.; Nayak, A.; Panwar, A.; and Basu, A. 2025. Pygraph: Robust compiler support for cuda graphs in pytorch. *arXiv preprint arXiv:2503.19779*.
- Gong, S.; Li, M.; Feng, J.; Wu, Z.; and Kong, L. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*.
- Gu, Y.; Dong, L.; Wei, F.; and Huang, M. 2023. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.
- Han, J.; Tian, M.; Zhu, J.; He, F.; Zhang, H.; Guo, S.; Zhu, D.; Tang, H.; Xu, P.; Guo, Y.; et al. 2025. Percept-WAM: Perception-enhanced world-awareness-action model for robust end-to-end autonomous driving. *arXiv preprint arXiv:2511.19221*.
- Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Hoogeboom, E.; Nielsen, D.; Jaini, P.; Forré, P.; and Welling, M. 2021. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in neural information processing systems*, 34: 12454–12465.
- Hu, Y.; Yang, J.; Chen, L.; Li, K.; Sima, C.; Zhu, X.; Chai, S.; Du, S.; Lin, T.; Wang, W.; et al. 2023. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 17853–17862.
- Hwang, J.-J.; Xu, R.; Lin, H.; Hung, W.-C.; Ji, J.; Choi, K.; Huang, D.; He, T.; Covington, P.; Sapp, B.; et al. 2024. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*.
- Jia, X.; Gao, Y.; Chen, L.; Yan, J.; Liu, P. L.; and Li, H. 2023a. Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7953–7963.
- Jia, X.; Wu, P.; Chen, L.; Xie, J.; He, C.; Yan, J.; and Li, H. 2023b. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21983–21994.
- Jia, X.; Yang, Z.; Li, Q.; Zhang, Z.; and Yan, J. 2024. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 37: 819–844.
- Jiang, B.; Chen, S.; Liao, B.; Zhang, X.; Yin, W.; Zhang, Q.; Huang, C.; Liu, W.; and Wang, X. 2024. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv preprint arXiv:2410.22313*.
- Jiang, B.; Chen, S.; Xu, Q.; Liao, B.; Chen, J.; Zhou, H.; Zhang, Q.; Liu, W.; Huang, C.; and Wang, X. 2023. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8340–8350.
- Kim, J.; Oh, J.; Yu, S.; Shin, H.; Kwak, D.; and Choi, J. W. 2026. SafeDrive: Fine-Grained Safety Reasoning for End-to-End Driving in a Sparse World. *arXiv preprint arXiv:2602.18887*.
- Kim, Y.; and Rush, A. M. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, 1317–1327.
- Li, B.; Zhang, Y.; Guo, D.; Zhang, R.; Li, F.; Zhang, H.; Zhang, K.; Zhang, P.; Li, Y.; Liu, Z.; et al. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.
- Li, J.; Wu, J.; Hu, D.; Huang, X.; Sun, B.; Hao, Z.; Lang, X.; Zhu, X.; and Zhang, L. 2026a. SGDrive: Scene-to-Goal Hierarchical World Cognition for Autonomous Driving. *arXiv preprint arXiv:2601.05640*.
- Li, P.; Zheng, Y.; Wang, Y.; Wang, H.; Zhao, H.; Liu, J.; Zhan, X.; Zhan, K.; and Lang, X. 2025a. Discrete diffusion for reflective vision-language-action models in autonomous driving. *arXiv preprint arXiv:2509.20109*.
- Li, X.; Thickstun, J.; Gulrajani, I.; Liang, P. S.; and Hashimoto, T. B. 2022. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35: 4328–4343.
- Li, Y.; Shang, S.; Liu, W.; Zhan, B.; Wang, H.; Wang, Y.; Chen, Y.; Wang, X.; An, Y.; Tang, C.; et al. 2025b. DriveVLA-WO: World models amplify data scaling law in autonomous driving. *arXiv preprint arXiv:2510.12796*.
- Li, Y.; Xiong, K.; Guo, X.; Li, F.; Yan, S.; Xu, G.; Zhou, L.; Chen, L.; Sun, H.; Wang, B.; et al. 2025c. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. *arXiv preprint arXiv:2506.08052*.
- Li, Y.; Zhou, L.; Yan, S.; Liao, B.; Yan, T.; Xiong, K.; Chen, L.; Xie, H.; Wang, B.; Chen, G.; et al. 2026b. UniDriveVLA: Unifying Understanding, Perception, and Action Planning for Autonomous Driving. *arXiv preprint arXiv:2604.02190*.
- Li, Y.; Zuo, Y.; He, B.; Zhang, J.; Xiao, C.; Qian, C.; Yu, T.; Gao, H.-a.; Yang, W.; Liu, Z.; et al. 2026c. Re-thinking On-Policy Distillation of Large Language Models: Phenomenology, Mechanism, and Recipe. *arXiv preprint arXiv:2604.13016*.
- Li, Z.; Li, K.; Wang, S.; Lan, S.; Yu, Z.; Ji, Y.; Li, Z.; Zhu, Z.; Kautz, J.; Wu, Z.; et al. 2024b. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*.
- Liang, Z.; Li, Y.; Yang, T.; Wu, C.; Mao, S.; Nian, T.; Pei, L.; Zhou, S.; Yang, X.; Pang, J.; et al. 2025. Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint arXiv:2508.20072*.
- Liao, B.; Chen, S.; Yin, H.; Jiang, B.; Wang, C.; Yan, S.; Zhang, X.; Li, X.; Zhang, Y.; Zhang, Q.; et al. 2025. Diffusion-drive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 12037–12047.
- Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

- Liu, Y.; Ding, P.; Jiang, T.; Wang, X.; Song, W.; Lin, M.; Zhao, H.; Zhang, H.; Zhuang, Z.; Zhao, W.; et al. 2026. MMaDA-VLA: Large Diffusion Vision-Language-Action Model with Unified Multi-Modal Instruction and Generation. *arXiv preprint arXiv:2603.25406*.
- Lou, A.; Meng, C.; and Ermon, S. 2023. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*.
- Ma, Y.; Cao, Y.; Sun, J.; Pavone, M.; and Xiao, C. 2024. Dolphins: Multimodal language model for driving. In *European Conference on Computer Vision*, 403–420. Springer.
- Mao, J.; Qian, Y.; Ye, J.; Zhao, H.; and Wang, Y. 2023. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415*.
- Marcu, A.-M.; Chen, L.; Hünermann, J.; Karnsund, A.; Hanotte, B.; Chidananda, P.; Nair, S.; Badrinarayanan, V.; Kendall, A.; Shotton, J.; et al. 2024. Lingoqa: Visual question answering for autonomous driving. In *European Conference on Computer Vision*, 252–269. Springer.
- Nie, M.; Peng, R.; Wang, C.; Cai, X.; Han, J.; Xu, H.; and Zhang, L. 2024. Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving. In *European Conference on Computer Vision*, 292–308. Springer.
- Nie, S.; Zhu, F.; You, Z.; Zhang, X.; Ou, J.; Hu, J.; Zhou, J.; Lin, Y.; Wen, J.-R.; and Li, C. 2025. Large Language Diffusion Models. *arXiv preprint arXiv:2502.09992*.
- Ou, J.; Nie, S.; Xue, K.; Zhu, F.; Sun, J.; Li, Z.; and Li, C. 2024. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*.
- Ranzato, M.; Chopra, S.; Auli, M.; and Zaremba, W. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.
- Renz, K.; Chen, L.; Arani, E.; and Sinavski, O. 2025. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 11993–12003.
- Sahoo, S. S.; Arriola, M.; Schiff, Y.; Gokaslan, A.; Marroquin, E.; Chiu, J. T.; Rush, A.; and Kuleshov, V. 2024. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37: 130136–130184.
- Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Shao, H.; Hu, Y.; Wang, L.; Song, G.; Waslander, S. L.; Liu, Y.; and Li, H. 2024. Lmdrive: Closed-loop end-to-end driving with large language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15120–15130.
- Shi, J.; Han, K.; Wang, Z.; Doucet, A.; and Titsias, M. 2024. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167.
- Sima, C.; Renz, K.; Chitta, K.; Chen, L.; Zhang, H.; Xie, C.; Beißwenger, J.; Luo, P.; Geiger, A.; and Li, H. 2024. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, 256–274. Springer.
- Singh, S.; Yadav, A.; Jain, J.; Shi, H.; Johnson, J.; and Desai, K. 2024. Benchmarking Object Detectors with COCO: A New Path Forward. *arXiv:2403.18819*.
- Song, K.; Sun, H.; Tan, X.; Qin, T.; Lu, J.; Liu, H.; and Liu, T.-Y. 2020. LightPAFF: A two-stage distillation framework for pre-training and fine-tuning. *arXiv preprint arXiv:2004.12817*.
- Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; and Liu, Y. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063.
- Tian, X.; Gu, J.; Li, B.; Liu, Y.; Wang, Y.; Zhao, Z.; Zhan, K.; Jia, P.; Lang, X.; and Zhao, H. 2024. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*.
- Tschannen, M.; Gritsenko, A.; Wang, X.; Naeem, M. F.; Alabdulmohsin, I.; Parthasarathy, N.; Evans, T.; Beyer, L.; Xia, Y.; Mustafa, B.; et al. 2025. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*.
- Von Rütte, D.; Fluri, J.; Ding, Y.; Orvieto, A.; Schölkopf, B.; and Hofmann, T. 2025. Generalized interpolating discrete diffusion. *arXiv preprint arXiv:2503.04482*.
- Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; et al. 2024. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.
- Wang, S.; Yu, Z.; Jiang, X.; Lan, S.; Shi, M.; Chang, N.; Kautz, J.; Li, Y.; and Alvarez, J. M. 2025a. Omnidrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In *Proceedings of the computer vision and pattern recognition conference*, 22442–22452.
- Wang, T.; Zhang, C.; Qu, X.; Li, K.; Liu, W.; and Huang, C. 2025b. Diffad: A unified diffusion modeling approach for autonomous driving. *arXiv preprint arXiv:2503.12170*.
- Wen, J.; Zhu, M.; Liu, J.; Liu, Z.; Yang, Y.; Zhang, L.; Zhang, S.; Zhu, Y.; and Xu, Y. 2025a. dvla: Diffusion vision-language-action model with multimodal chain-of-thought. *arXiv preprint arXiv:2509.25681*.
- Wen, Y.; Li, H.; Gu, K.; Zhao, Y.; Wang, T.; and Sun, X. 2025b. Llada-vla: Vision language diffusion action models. *arXiv preprint arXiv:2509.06932*.
- Wu, P.; Jia, X.; Chen, L.; Yan, J.; Li, H.; and Qiao, Y. 2022. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 35: 6119–6132.
- Xie, S.; Kong, L.; Dong, Y.; Sima, C.; Zhang, W.; Chen, Q. A.; Liu, Z.; and Pan, L. 2025. Are VLMs Ready for Autonomous Driving? An Empirical Study from the Reliability, Data and Metric Perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6585–6597.
- Xu, M.; Cui, J.; Cai, F.; Shang, H.; Zhu, Z.; Luan, S.; Xu, Y.; Zhang, N.; Li, Y.; Cai, J.; et al. 2025a. WAM-Diff: A Masked Diffusion VLA Framework with MoE and Online Reinforcement Learning for Autonomous Driving. *arXiv preprint arXiv:2512.11872*.

- Xu, Z.; Bai, Y.; Zhang, Y.; Li, Z.; Xia, F.; Wong, K.-Y. K.; Wang, J.; and Zhao, H. 2025b. Drivegpt4-v2: Harnessing large language model capabilities for enhanced closed-loop autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 17261–17270.
- Xu, Z.; Zhang, Y.; Xie, E.; Zhao, Z.; Guo, Y.; Wong, K.-Y. K.; Li, Z.; and Zhao, H. 2024. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 9(10): 8186–8193.
- Yang, L.; Tian, Y.; Li, B.; Zhang, X.; Shen, K.; Tong, Y.; and Wang, M. 2025a. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*.
- Yang, Z.; Chai, Y.; Jia, X.; Li, Q.; Shao, Y.; Zhu, X.; Su, H.; and Yan, J. 2025b. DriveMoE: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. *arXiv preprint arXiv:2505.16278*.
- Yang, Z.; Chai, Y.; Jia, X.; Li, Q.; Shao, Y.; Zhu, X.; Su, H.; and Yan, J. 2026. Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10678–10688.
- Yao, W.; Li, Z.; Lan, S.; Wang, Z.; Sun, X.; Alvarez, J. M.; and Wu, Z. 2026. Drivesuprim: Towards precise trajectory selection for end-to-end planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 11910–11918.
- Ye, J.; Xie, Z.; Zheng, L.; Gao, J.; Wu, Z.; Jiang, X.; Li, Z.; and Kong, L. 2025a. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*.
- Ye, Z.; Chen, L.; Lai, R.; Lin, W.; Zhang, Y.; Wang, S.; Chen, T.; Kasikci, B.; Grover, V.; Krishnamurthy, A.; et al. 2025b. Flashinfer: Efficient and customizable attention engine for llm inference serving. *Proceedings of Machine Learning and Systems*, 7.
- You, Z.; Nie, S.; Zhang, X.; Hu, J.; Zhou, J.; Lu, Z.; Wen, J.-R.; and Li, C. 2025. LLaDA-V: Large Language Diffusion Models with Visual Instruction Tuning. *arXiv preprint arXiv:2505.16933*.
- Zhai, J.-T.; Feng, Z.; Du, J.; Mao, Y.; Liu, J.-J.; Tan, Z.; Zhang, Y.; Ye, X.; and Wang, J. 2023. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint arXiv:2305.10430*.
- Zhang, J.; Gagvani, M.; Cui, C.; Peng, J.; Zhang, R.; and Wang, Z. 2026. Efficient and Explainable End-to-End Autonomous Driving via Masked Vision-Language-Action Diffusion. *arXiv preprint arXiv:2602.20577*.
- Zheng, Y.; Liang, R.; Zheng, K.; Zheng, J.; Mao, L.; Li, J.; Gu, W.; Ai, R.; Li, S. E.; Zhan, X.; et al. 2025. Diffusion-based planning for autonomous driving with flexible guidance. *arXiv preprint arXiv:2501.15564*.
- Zheng, Z.; Chen, S.; Yin, H.; Zhang, X.; Zou, J.; Wang, X.; Zhang, Q.; and Zhang, L. 2026. Resad: Normalized residual trajectory modeling for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3729–3739.
- Zhou, X.; Han, X.; Yang, F.; Ma, Y.; and Knoll, A. C. 2025a. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. *arXiv preprint arXiv:2503.23463*.
- Zhou, Z.; Cai, T.; Zhao, S. Z.; Zhang, Y.; Huang, Z.; Zhou, B.; and Ma, J. 2025b. AutoVLA: A Vision-Language-Action Model for End-to-End Autonomous Driving with Adaptive Reasoning and Reinforcement Fine-Tuning. *arXiv preprint arXiv:2506.13757*.
- Zhu, J.; Wang, W.; Chen, Z.; Liu, Z.; Ye, S.; Gu, L.; Tian, H.; Duan, Y.; Su, W.; Shao, J.; Gao, Z.; Cui, E.; Wang, X.; Cao, Y.; Liu, Y.; Wei, X.; Zhang, H.; Wang, H.; Xu, W.; Li, H.; Wang, J.; Deng, N.; Li, S.; He, Y.; Jiang, T.; Luo, J.; Wang, Y.; He, C.; Shi, B.; Zhang, X.; Shao, W.; He, J.; Xiong, Y.; Qu, W.; Sun, P.; Jiao, P.; Lv, H.; Wu, L.; Zhang, K.; Deng, H.; Ge, J.; Chen, K.; Wang, L.; Dou, M.; Lu, L.; Zhu, X.; Lu, T.; Lin, D.; Qiao, Y.; Dai, J.; and Wang, W. 2025. InternVL3: Exploring Advanced Training and Test-Time Recipes for Open-Source Multimodal Models. *arXiv:2504.10479*.
- Zou, J.; Chen, S.; Liao, B.; Zheng, Z.; Song, Y.; Zhang, L.; Zhang, Q.; Liu, W.; and Wang, X. 2025. Diffusion-DriveV2: Reinforcement Learning-Constrained Truncated Diffusion Modeling in End-to-End Autonomous Driving. *arXiv:2512.07745*.

A Appendix

A.1 Training Datasets Construction

To construct a unified, multi-task driving corpus, we aggregate and standardize multimodal data from five distinct sources. The resulting consolidated dataset comprises 981,361 samples for general multi-task learning, alongside dedicated splits for task-specific evaluations. Every sample is mapped to a standardized conversation format: an input containing visual streams and language context, and a target sequence representing actions, text, or structured coordinates.

- NAVSIM (Dauner et al. 2024): We utilize the official “Navtest” training split containing 103,288 samples. The inputs consist of multi-view camera streams, navigation targets, and ego-telemetry (velocity and acceleration), mapped to an 8-waypoint trajectory optimization target.
- LingoQA (Marcu et al. 2024): To embed robust natural-language scene-reasoning and temporal video understanding, we incorporate 413,829 samples. Each instance has five sequentially ordered driving frames associated with an open-ended question regarding situational behaviors.
- DriveLM (Sima et al. 2024): We integrate 346,978 question-answering pairs across 4,072 frames to structurally enforce logical dependencies among perception, prediction, and planning tasks.
- COCO (Singh et al. 2024): General spatial grounding and cross-modal object localization capabilities are preserved by incorporating 117,266 samples converted into a referring expression format.
- SimLingo (Bench2Drive) (Renz et al. 2025): For closed-loop reactive planning, we utilize all 2,024,407 simulation frames from Bench2Drive. The configuration requires generating 6 future waypoints conditioned on structural camera tokens and ego-states.

Table 8: Overview of the datasets used for training.

Dataset	Domain	# Samples	Task
<i>Unified Multi-Task Split</i>			
NAVSIM	Driving	103,288	Planning
DriveLM	Driving	346,978	Driving VQA
LingoQA	Driving	413,829	Driving VQA
COCO-D	Vision	117,266	Detection
<i>Task-Specific Splits</i>			
SimLingo	Driving	2,024,407	Closed-loop planning
NAVSIM	Driving	103,288	Open-loop planning

A.2 Quantitative Evaluation Metrics

To rigorously validate the multi-task and closed-loop capabilities of WAM-Diff2, we evaluate performance across five standard benchmarks spanning open/closed-loop motion planning, driving scene understanding, and spatial grounding.

Open-Loop Motion Planning. On the NAVSIM v1 benchmark, we report the official Predictive Driver Model Score (PDMS). The PDMS provides a holistic metric by evaluating a compound score of no at-fault collision (NC), drivable area compliance (DAC), time-to-collision (TTC), driving comfort (Comf.), and ego progress (EP):

$$\text{PDMS} = \text{NC} \cdot \text{DAC} \cdot \frac{5 \text{ EP} + 5 \text{ TTC} + 2 \text{ Comf}}{12}. \quad (9)$$

For the NAVSIM v2 benchmark, we adopt the Extended Predictive Driver Model Score (EPDMS). This metric introduces a broader set of constraints, including driving direction compliance (DDC), traffic light compliance (TLC), lane keeping (LK), history comfort (HC), and extended comfort (EC):

$$\text{EPDMS} = \text{NC} \cdot \text{DAC} \cdot \text{DDC} \cdot \text{TLC} \cdot \frac{5 \text{ EP} + 5 \text{ TTC} + 2 \text{ LK} + 2 \text{ HC} + 2 \text{ EC}}{16}. \quad (10)$$

Closed-Loop Reactive Control. On the CARLA-driven Bench2Drive benchmark, we report the Success Rate (SR) and Driving Score (DS) to assess reactive planning under dynamic distribution shifts. Let N_{total} be the total number of evaluated routes and N_{success} be the number of successfully completed routes. The metrics are formulated as follows:

$$\text{SR} = \frac{N_{\text{success}}}{N_{\text{total}}}, \quad (11)$$

$$\text{DS} = \frac{1}{N_{\text{total}}} \sum_{i=1}^{N_{\text{total}}} \left(\text{RC}_i \times \prod_{j=1}^{M_i} \text{IS}_{i,j} \right), \quad (12)$$

where $\text{RC}_i \in [0, 100]$ represents the completion percentage of the i -th route, and $\text{IS}_{i,j} \in [0, 1)$ denotes the multiplicative penalty multiplier for the j -th infraction committed along that route.

Scene Reasoning and Spatial Grounding. For driving-oriented VQA, we evaluate open-ended situational scene reasoning via the Lingo-Judge metric on the LingoQA dataset and the GPT Score on the DriveBench dataset. As for visual perception, To quantify general cross-modal object localization and spatial grounding proficiency without task-specific heads, we evaluate structural JSON token predictions on the COCO dataset using standard mean Average Precision (mAP) at IoU thresholds from 0.50 to 0.95.

A.3 Multi-Stage Optimization Protocol

Model Architecture and Preprocessing. The framework is built upon the pre-trained Qwen3-VL-2B vision-language backbone. Multi-view camera inputs are uniformly scaled to 1920×1080 pixels and segmented into tokens using a spatial patch size of 16×16 . The resulting visual embeddings are projected into the language model’s hidden dimension space and concatenated with textual instruction tokens to establish a unified multimodal representation for downstream generation.

Training Phases and Objectives. The transformation from a sequential autoregressive generalist to a parallel discrete diffusion agent proceeds via a three-phase optimization pipeline:

Phase 1: Multi-Task Autoregressive Pretraining. To establish a robust semantic foundation, the base model undergoes supervised fine-tuning for 5 epochs using a standard next-token prediction objective. All parameters are jointly optimized using the AdamW optimizer with a weight decay of 0.05, a cosine learning rate schedule, and a 10% linear warmup. The base learning rate is fixed at 4×10^{-5} under a global batch size of 128.

Phase 2: Progressive Block-Causal Adaptation. To manage the severe architectural shift from sequential to parallel generation, we incrementally relax causal constraints by scaling the decoding block size:

$$\text{AR} \rightarrow B_4 \rightarrow B_8 \rightarrow B_{16} \rightarrow B_{32}. \quad (13)$$

Each block variant is initialized with the weights of its predecessor ($\theta_B^{(0)} \leftarrow \theta_{B/2}$). For each configuration, the network is optimized for 5 epochs with a global batch size of 128. To stabilize cross-modal alignment during the attention shift, we apply a decoupled learning rate strategy: the language backbone is maintained at 4×10^{-5} , while the vision encoder is restricted to 2×10^{-6} .

Phase 3: Hierarchical Knowledge Distillation. To eliminate exposure bias and close the capacity gap resulting from compact parameterization, the 2B block-32 student model undergoes dual-stage distillation for 5 epochs. First, it is aligned with a stable 2B block-4 diffusion teacher (Stage II). Second, it inherits advanced cognitive reasoning from an 8B block-32 diffusion teacher (Stage III). Both stages optimize a symmetric Jensen-Shannon Divergence (JSD) loss over intermediate noisy states using a unified learning rate of 2×10^{-6} across all components.

Hardware Infrastructure. The framework is implemented in PyTorch. Large-scale distributed training is executed across four compute nodes, each equipped with Kunpeng 920 7285Z processors and eight Ascend 910C NPUs., requiring a total allocation detailed in Table 9. Comprehensive inference throughput and closed-loop evaluations are performed on a single node utilizing eight Ascend 910C NPUs. Hardware-specific inference optimizations are evaluated separately from the main experiments. In particular, the FlashInfer and CUDA Graph implementations are assessed only in the system-level efficiency study on a CUDA-compatible platform, whereas all model-quality results are obtained on the Ascend platform.

A.4 Multi-Task Prompt Templates

We formulate all tasks using a unified vision-language instruction-following format. Each training sample is converted into a conversation consisting of a user message and an assistant response. The user message contains the visual input, task-specific context, and an explicit instruction, while the assistant response is used as the prediction target. We use the same prompt templates during training and evaluation, unless otherwise specified.

In the following templates, `<image>` denotes the visual placeholder processed by the vision encoder, and the contents

Table 9: Stage-wise training hyperparameters.

Hyperparameter	Stage I	Stage II	Stage III
Epochs	5	5	5
Batch size	128	128	128
Learning rate	4×10^{-5}	2×10^{-6}	2×10^{-6}
Weight decay	0.05	0.05	0.05
Warmup ratio	0.1	0.1	0.1
LR schedule	Cosine	Cosine	Cosine
NPU hours	1,376	240	264

enclosed by angle brackets denote sample-specific fields. Although the input prompts differ across tasks, all tasks share the same autoregressive or block-diffusion language modeling interface.

Motion Planning. For NAVSIM, the model receives multi-view images together with navigation and ego-state information, including the current position, velocity, and acceleration. It is instructed to predict eight future waypoints covering the next four seconds.

```
Here is front-view image from a driving
vehicle: <image><image><image>
The navigation information is:
<navigation>.
The current position is (<position_x>,
<position_y>).
The current velocity is: (<velocity_x>,
<velocity_y>) and the current
acceleration is: (<acceleration_x>,
<acceleration_y>).
Instruction: Based on the visual
motion cues from the image (such as
the relative speed of other vehicles
and the changing distance to the
intersection) and the provided
telemetry, predict the optimal driving
action for the next 4 seconds with 8
new waypoints.
```

The target response is represented as a flattened sequence of two-dimensional coordinates:

```
<x_1>, <y_1>, <x_2>, <y_2>, ..., <x_8>, <y_8>
```

Similarly, for Bench2Drive, the model takes a front-view image, navigation targets (as a sequence of coordinates), a high-level command (e.g., "follow the road"), and the current scalar velocity. It is instructed to predict six future waypoints covering the next three seconds.

```
Here is a front-view image from a
driving vehicle:
<image>
The navigation information is:
Target: [<target_1_x>, <target_1_y>]
[<target_2_x>, <target_2_y>]. Command:
<command>
Current velocity is: <current_velocity>
Predict the optimal driving action
for the next 3 seconds with 6 new
waypoints.
```

The target response follows the same flattened sequence structure, corresponding to the 6 predicted waypoints:

```
<x_1>, <y_1>, <x_2>, <y_2>, ..., <x_6>, <y_6>
```

No additional natural-language explanation is included in the planning target.

Driving Question Answering. We retain the original dataset-specific prompt formats for LingoQA and DriveBench. For LingoQA, each sample consists of five temporally ordered driving frames followed by the original natural-language question:

```
<image> <image> <image> <image> <image>
<question>
```

For DriveBench, the input may contain either a single camera view or six surround-view images, with each image preceded by its camera identifier:

```
[CAM_VIEW] <image> ...
<question>
```

DriveBench includes heterogeneous question types, such as action reasoning, collision-risk analysis, object-referenced reasoning, and multiple-choice ego-behavior prediction. We preserve the original questions, camera identifiers, object references, and answer annotations for both datasets without rewriting. The model directly generates the corresponding natural-language answer or selected option.

Visual Grounding. For COCO visual grounding, the model receives an image together with a referring expression using the following prompt:

```
<image>
Locate every object that matches the
description
"{ref_sentence}" in the image. Report
bbox
coordinates in JSON format.
```

The model is required to locate all objects matching the referring expression and return their bounding boxes and semantic labels in JSON format. Each detected object is represented by a `bbox_2d` field and a `label` field. The bounding box is serialized in the coordinate order $[y_{\min}, x_{\min}, y_{\max}, x_{\max}]$, where all coordinates are normalized to a 1000×1000 coordinate space. The target response follows the format:

```
[
  {
    "bbox_2d": [y_min, x_min, y_max, x_max],
    "label": "<object_class>"
  },
  ...
]
```

The output contains only the JSON-formatted detection results without any additional explanatory text.

Unified Output Interface. Despite the differences in task semantics, all targets are represented as token sequences and optimized with the same language-modeling objective. Motion planning and visual grounding use strictly formatted numerical outputs, whereas driving question-answering and scene-understanding tasks use natural-language outputs.

During multi-task training, task identity is specified implicitly through the task-specific instruction and output format, without introducing additional task-specific prediction heads.

A.5 Additional Qualitative Results

We present extensive qualitative visualizations of WAMDiff2 on NAVSIM and Bench2Drive to demonstrate the effectiveness of our proposed method (see Figure 12 and 13).



Figure 12: Representative extreme-weather driving scenarios on Bench2Drive, specifically featuring Night & Fog, Dense Fog, and Rain & Fog conditions.

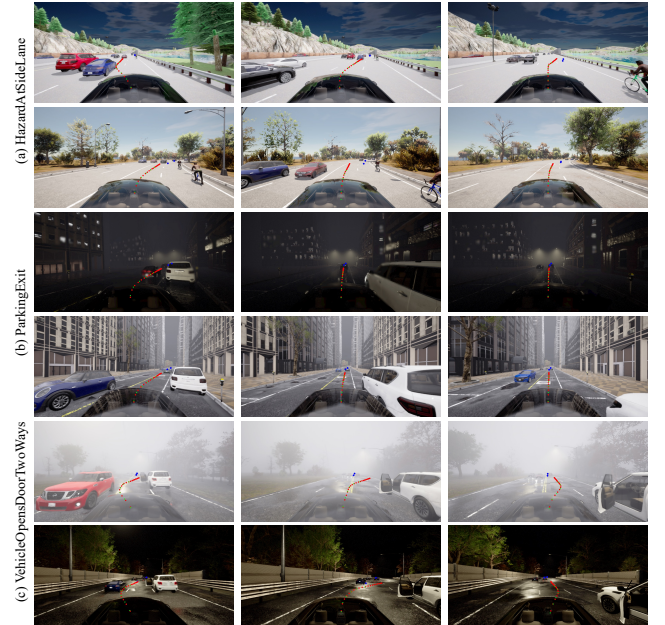


Figure 13: Representative driving scenarios in Bench2Drive under diverse weather and lighting conditions, including (a) HazardAtSideLane, (b) ParkingExit, and (c) VehicleOpensDoorTwoWays