WAM-DIFF2: HIERARCHICAL AR-TO-DIFFUSION DISTILLATION FOR HIGHLY EFFICIENT AUTONOMOUS DRIVING VLA

Anonymous authors
Paper under double-blind review

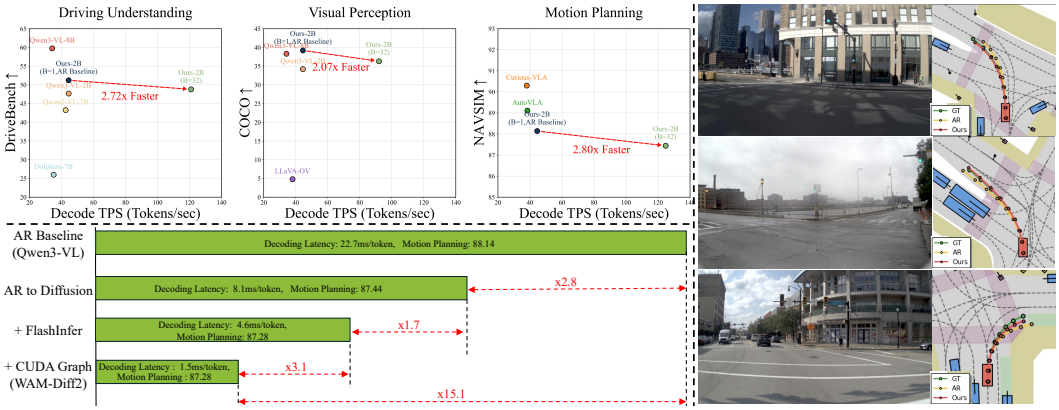


Figure 1: Overview of WAM-Diff2. **Top left**: with $B = 32$, WAM-Diff2 performs comparably to the AR baseline on DriveBench, COCO, and NAVSIM while decoding 2.1–2.8× faster. **Bottom left**: on NAVSIM, the AR-to-diffusion transition alone yields a 2.8× speedup, which FlashInfer and CUDA Graphs raise to 15.1×. **Right**: WAM-Diff2 alleviates the compounding drift of AR decoding in long-horizon trajectory prediction.

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 induced by teacher-forced training. 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 scene understanding, visual perception, and motion planning benchmarks demonstrate that WAM-Diff2 effectively mitigates long-horizon error accumulation and retains multi-task capabilities with bounded performance degradation relative to autoregressive baselines. Crucially, the autoregressive-to-diffusion transition yields a 2.8× decoding speedup, which scales to an ultimate 15.1× acceleration when combined with system-level optimizations including FlashInfer and CUDA Graphs.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Vision-Language-Action models (Sima et al., 2024; Hwang et al., 2024; Li et al., 2025b; Chen et al., 2025; Li et al., 2025c; Han et al., 2025; Li et al., 2025a; Cui et al., 2025; Zhou et al., 2025a;b; Li et al., 2026c) 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., 2026c; 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 & 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 et al., 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 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, easing the attention shift. In Stage II (Block-Wise Distillation), we actively expose the student to intermediate noisy states sampled from its own parallel decoding, and perform on-policy distillation (Agarwal et al., 2024; Gu et al., 2023) from a 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 near-parity on planning with bounded degradation on perception and scene reasoning relative to its autoregressive foundation, while delivering a substantial

decoding 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. Overall, our framework offers a scalable approach to converting pretrained autoregressive VLAs into high-throughput diffusion models, supporting efficient deployment while largely preserving their multitask capabilities.

2 Related Works

VLA for Autonomous Driving. Vision-Language-Action (VLA) models for driving have progressed from 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) to 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). Recent driving generalists such as EMMA (Hwang et al., 2024), Percept-WAM (Han et al., 2025), DriveMoE (Yang et al., 2026), and UniDriveVLA (Li et al., 2026c) integrate perception and planning within autoregressive frameworks (Mao et al., 2023; Chen et al., 2025; Zhou et al., 2025a;b; Li et al., 2026c), and are therefore subject to unidirectional causal constraints, high inference latency, and exposure bias. Speculative decoding with parallel block drafters (Chen et al., 2026a; Cheng et al., 2026b), recently applied to driving VLAs (Li et al., 2026d), reduces this latency but introduces an additionally trained drafter; moreover, since it preserves the output distribution of the autoregressive target, exposure bias remains. In contrast, we convert the VLA itself into a block-diffusion model that decodes in parallel without additional models.

Discrete Diffusion Models. Discrete diffusion models generate sequences through iterative token denoising (Austin et al., 2021; Hoogeboom et al., 2021; Campbell et al., 2022; Li et al., 2022; Gong et al., 2022; Lou et al., 2023). Masked diffusion (Shi et al., 2024; Sahoo et al., 2024; Ou et al., 2024; Gat et al., 2024; Von Rütte et al., 2025) scales this formulation to language (Nie et al., 2025; Ye et al., 2025a) and multimodal generation (You et al., 2025; Yang et al., 2025), with approximate KV caching and confidence-aware parallel decoding improving inference efficiency (Wu et al., 2025). Block diffusion (Arriola et al., 2025) combines autoregressive generation across blocks with parallel denoising within blocks, providing a route to adapting pretrained autoregressive models. SDAR (Cheng et al., 2025) and Fast-dLLM v2 (Wu et al., 2026b) pursue this conversion through continued fine-tuning of language models; SDAR-VL (Cheng et al., 2026a) addresses training stability for vision-language understanding, while Fast-dVLM (Wu et al., 2026a) converts autoregressive VLMs through block-size annealing. A related line applies discrete diffusion to robotic actions and driving trajectories (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;a; Liu et al., 2026; Chen et al., 2026b). Our work connects these directions by studying the conversion of a unified autoregressive VLA to discrete diffusion, with the goal of preserving its multitask scope beyond action and trajectory generation.

LLM and LVM Distillation. Knowledge distillation (Hinton et al., 2015) is critical for model compression. Traditional off-policy distillation for autoregressive models (Kim & 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., 2026b; Afsharrad et al., 2026) and divergence-based objectives (Gu et al., 2023) seek to mitigate. However, transferring knowledge from autoregressive teachers to diffusion students is uniquely challenging due to the discrepancy between causal and bidirectional attention patterns. Unlike standard diffusion trajectory distillation, our tiered framework addresses this cross-paradigm gap through noise-conditioned distribution matching across progressive block-causal adaptations and cross-scale architectures.

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).

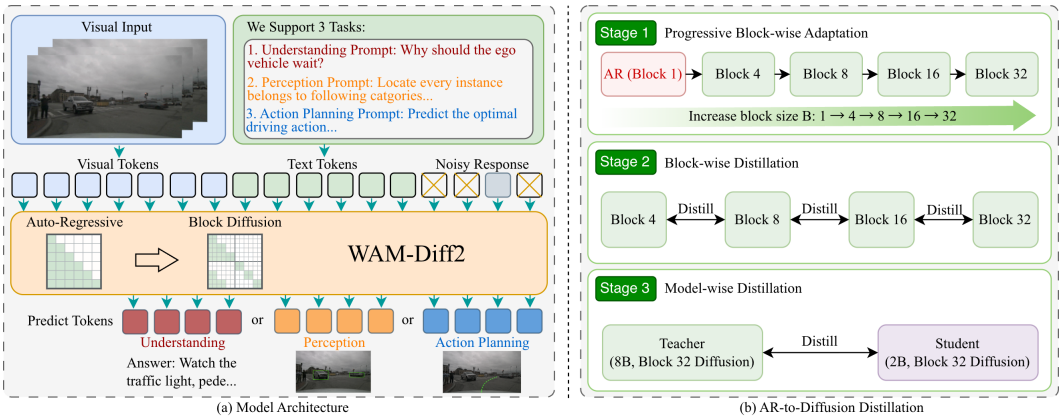


Figure 2: Architectural overview of WAM-Diff2. (a) Multi-view images, task instructions, and noisy response tokens are processed by a single model for driving understanding, perception, and planning; replacing the causal mask with a block-causal mask enables parallel denoising within each block while preserving causality across blocks. (b) AR-to-diffusion transformation: Stage 1 progressively enlarges the block size ($B = 1 \rightarrow 32$); Stage 2 distills each block-B model from the block-$B/2$ model; Stage 3 (optional) distills an 8B block-32 diffusion teacher into the 2B student.

3.1 Preliminaries

VLA Formulation. A multi-task driving VLA Φ_θ takes as input a set of camera images $\mathcal{I}^{\text{cam}}$ (multi-view and/or multi-frame) and a task instruction $\mathcal{L}$; for planning, $\mathcal{L}$ additionally contains the navigation command and the ego status (e.g., velocity and acceleration). It covers three tasks: *driving understanding*, which answers a scene-related question in natural language $\mathcal{T}^{\text{qa}}$; *visual perception*, which localizes the queried objects as labeled 2D boxes $\mathcal{T}^{\text{det}} = \{(c_j, \mathbf{b}_j)\}_{j=1}^{M}$ with $\mathbf{b}_j \in \mathbb{R}^4$; and *motion planning*, which predicts H future BEV waypoints $\mathcal{T}^{\text{traj}} = \{\mathbf{w}_h\}_{h=1}^{H}$ with $\mathbf{w}_h \in \mathbb{R}^2$. Following recent driving VLAs (Hwang et al., 2024; Li et al., 2025c), all outputs are serialized as text over a shared vocabulary $\mathcal{V}$ without task-specific heads (Appendix A.4), i.e., $\mathbf{y} = (y_1, \ldots, y_L) = \Phi_\theta(x)$ with $x = (\mathcal{I}^{\text{cam}}, \mathcal{L})$, where $\mathbf{y}$ is the serialized form of $\mathcal{T}^{\text{qa}}$, $\mathcal{T}^{\text{det}}$, or $\mathcal{T}^{\text{traj}}$ depending on the task.

Autoregressive VLA. Conventional driving generalists implement Φ_θ with a causal factorization $p_\theta(\mathbf{y} \mid x) = \prod_{i=1}^{L} p_\theta(y_i \mid \mathbf{y}_{<i}, x)$, so inference requires L sequential forward passes. Because training conditions on ground-truth prefixes (teacher forcing), errors in the model's own early predictions propagate to later tokens at inference, a phenomenon known as exposure bias.

Discrete Diffusion VLA. A discrete diffusion VLA instead generates $\mathbf{y}$ by iterative parallel denoising. We adopt absorbing-state (masked) diffusion (Austin et al., 2021; Sahoo et al., 2024; Shi et al., 2024): the forward process independently replaces each token of $\mathbf{y}$ with [MASK] with probability $t \in [0, 1]$, yielding $\mathbf{y}^t$, and a denoiser $p_\theta(y_i \mid \mathbf{y}^t, x)$ predicts all masked tokens in parallel. Following block diffusion (Arriola et al., 2025), we partition $\mathbf{y}$ into non-overlapping blocks of size B with a block-causal attention mask, so blocks are generated autoregressively while tokens within a block are denoised jointly under bidirectional attention; the AR VLA is the special case $B = 1$. At inference, each block starts fully masked; at each step, all masked tokens with confidence above a threshold τ (default 0.9) are unmasked with their argmax predictions, together with at least the most confident one, and the rest remain masked. Each block thus takes an adaptive number of steps, at most B, and an optional budget $S_{\max}$ bounds the worst-case latency by unmasking all remaining tokens at the final step. The number of sequential forward passes is therefore at most $\lceil L/B \rceil \cdot \min(B, S_{\max})$, versus L for AR decoding.

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, 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: starting from the AR model ($B = 1$), we progressively enlarge the block size $B \in \{4, 8, 16, 32\}$. For each B, a masking level t is drawn independently for each block $\mathcal{B}$ via stratified sampling to obtain the noisy block state $\mathbf{y}_{\mathcal{B}}^t$, and p_{θ_B} is trained to reconstruct the tokens of $\mathcal{B}$ given x, the clean context $\mathbf{y}_{\setminus\mathcal{B}}$, $\mathbf{y}_{\mathcal{B}}^t$, and t:

$$\mathcal{L}_{\mathrm{SFT}}^{B} = -\mathbb{E}_{t,\mathcal{B}:|\mathcal{B}|=B,\mathbf{y}_{\mathcal{B}}^t} \sum_{i\in\mathcal{B}} \log p_{\theta_B}(y_i \mid x, \mathbf{y}_{\setminus\mathcal{B}}, \mathbf{y}_{\mathcal{B}}^t, t). \tag{1}$$

To prevent semantic degradation during the attention shift, each block size is initialized from the preceding one, $\theta_B^{(0)} \leftarrow \theta_{B/2}^{\mathrm{SFT}}$, with the block-4 model initialized from θ_{AR}. 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 $4 \to 8 \to 16 \to 32$. 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 intermediate states sampled from the student's own parallel decoding, so that distillation is performed on-policy. Motivated by the concern that the mode-covering behavior of the standard forward KL divergence may blur multi-modal trajectory distributions, we align the output distributions across the target block B using a symmetric Jensen-Shannon Divergence (JSD) loss (Englesson & Azizpour, 2021):

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

where $\tilde{\mathbf{y}}_t^B$ denotes an intermediate noisy state sampled from the student's own decoding trajectory, and the JSD between two distributions p and q is defined as:

$$\mathrm{JSD}(p \parallel q) = \frac{1}{2}\mathrm{KL}\left(p \parallel \frac{p+q}{2}\right) + \frac{1}{2}\mathrm{KL}\left(q \parallel \frac{p+q}{2}\right). \tag{3}$$

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.

Stage III: Model-Wise Cross-Scale Distillation. Deployment favors compact models, yet smaller backbones are generally less capable. We therefore transfer knowledge from an 8B diffusion teacher to the 2B student, improving the student while preserving its decoding cost. We observe that knowledge transfer is most effective when the teacher and student share consistent "thinking patterns", which we quantify by the top-K overlap ratio $\rho_K(p_T, p_S) = |\mathrm{TopK}(p_T) \cap \mathrm{TopK}(p_S)|/K$ between the teacher and student predictive distributions p_T and p_S. Our empirical analysis shows that ρ_K is consistently higher between two diffusion models than between an autoregressive teacher and a diffusion student, with the gap most pronounced for $K \geq 5$. Therefore, we utilize an 8B block-32 diffusion model $p_{\phi_{32}}^{8B}$ as our ultimate teacher. The 2B block-32 student is trained on its own decoding trajectories with the same objective as Eq. (2), with the teacher $p_{\theta_{B/2}}$ replaced by $p_{\phi_{32}}^{8B}$; we denote this loss $\mathcal{L}_{\mathrm{MWD}}$. 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.

4 Experiments

4.1 Experimental Setups

Implementation Details. WAM-Diff2 is instantiated at 2B and 8B scales on Qwen3-VL (Bai et al., 2025a), reusing its vision encoder, cross-modal adapter, and language backbone; the only architectural change replaces the causal attention mask with the block-causal mask of Section 3.1, so no parameters

Table 1: Multi-task performance and decoding efficiency under the unified multi-task protocol (Section 4.1), where each model is a single checkpoint evaluated on all tasks. Metrics: GPT Score (DriveBench), Lingo-Judge (LingoQA), mAP (COCO), PDMS (NAVSIM v1), and EPDMS (NAVSIM v2). Decode TPS is measured on identical hardware; for our diffusion variants ($B \geq 4$), the two values are without / with FlashInfer and CUDA Graphs. †: reinforcement learning; *: score-based candidate selection. **Bold** and underline denote the best and second-best results in each column; the two values in a cell are ranked separately. The AR baseline ($B = 1$) shares the backbone and training data with our diffusion variants.

Method	Driving Understanding		Visual Perception	Motion Planning		Efficiency
	DriveBench ↑	LingoQA ↑	COCO ↑	NAVSIM V1 ↑	NAVSIM V2 ↑	Decode TPS ↑
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)	56.71	67.20	–	86.50 / 90.80†	–	–
Ours-2B ($B = 1$, AR Baseline)	51.23	**68.40**	39.20	**88.14 / 91.50***	**88.32**	44.5
Ours-2B ($B = 4$)	48.92	68.20	37.80	87.87 / 91.47*	87.99	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	108.4 / 635.3
Ours-2B ($B = 32$)	48.80	65.80	36.30	87.44 / 91.05*	87.50	**124.8 / 673.4**

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	91.1	99.1	98.0	99.4	99.8	88.0	98.5	97.2	98.2	86.4	**90.7**

are added. The AR baseline is first obtained by multi-task next-token-prediction training; this phase can be skipped when a pre-trained AR driving generalist is available. The unified multi-task model is trained on NAVSIM, DriveLM (Sima et al., 2024), LingoQA, and COCO, and the task-specific planners on NAVSIM or SimLingo (Renz et al., 2025) for Bench2Drive (Table 9). Each training phase (the AR baseline, each block size in Stages I and II, and Stage III) runs for 5 epochs with AdamW (weight decay 0.05), a cosine schedule with 10% linear warmup, and a global batch size of 128 on 32 Ascend 910C NPUs; stage-wise learning rates are given in Appendix A.3. Multi-view camera inputs are resized to 1920×1080. Model-quality results are obtained on Ascend NPUs, whereas decoding throughput is measured on a CUDA-compatible platform.

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,

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 ↑				Multi-Ability Test (%) ↑					
	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-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-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	69.09	20.00	71.34	42.03
Orion (Fu et al., 2025)	151.48	17.38	**54.62**	77.74	25.00	71.11	**78.33**	33.00	69.15	54.72
DriveMoE (Yang et al., 2026)	175.96	15.31	48.64	74.22	34.67	40.00	65.45	40.00	59.44	47.91
UniDriveVLA (Li et al., 2026c)	**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	80.01	**47.50**	60.00	60.00	**50.00**	88.42	61.18
Ours-2B (B=8)	118.17	19.26	50.23	79.73	46.25	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

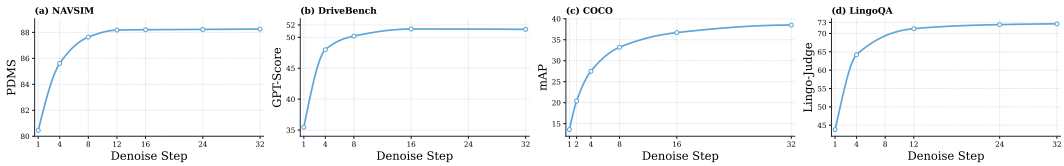


Figure 3: Effect of the maximum number of denoising steps $S_{\max}$ under confidence-based decoding ($\tau = 0.9$, $B = 32$). Performance improves as $S_{\max}$ increases.

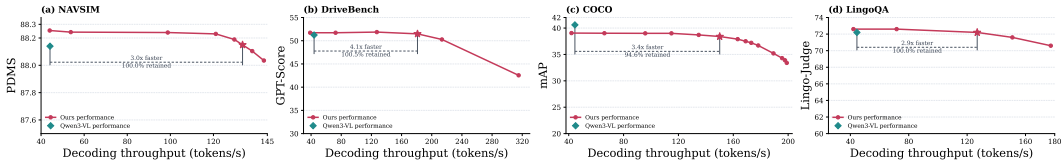


Figure 4: Performance–efficiency trade-offs of WAM-Diff2 across multiple benchmarks, obtained by varying the confidence threshold τ at inference; a lower τ unmasks more tokens per step and yields higher decoding throughput.

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 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, retaining over 95% of the semantic reasoning capabilities relative to its autoregressive baseline sharing the same backbone. For spatial perception, the model maintains competitive visual localization with 36.3 mAP on COCO, showing that dense cross-modal alignment remains largely 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 effectively preserves holistic scene reasoning and planning precision with bounded degradation while unlocking substantial gains in decoding efficiency.

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	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 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.

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 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	ρ_1 ↑	ρ_5 ↑	ρ_{10} ↑	ρ_{20} ↑
AR Teacher	83.6	51.2	54.3	46.1
Diffusion Teacher	**84.8**	**58.2**	**60.5**	**50.7**

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**

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 attaining a Success Rate of 49.55% and a Driving Score of 78.93. Corresponding trajectory rollouts are visualized in Figure 5 and Appendix Figure 8.

Inference Efficiency. Parallel block-wise refinement is the source of the speedup. Without system-level optimizations, decoding throughput rises from 44.5 TPS for the AR baseline to 68.3–124.8 TPS as B grows from 4 to 32 (1.5–2.8×; Table 1), while PDMS and EPDMS on NAVSIM drop by at most 0.8 and 1.0 points relative to the AR baseline across block sizes; open-ended VQA and grounding are more sensitive, dropping by up to 2.5 on DriveBench, 2.6 on LingoQA, and 2.9 mAP on COCO. For a given model, the confidence threshold τ and the step budget $S_{\max}$ can be adjusted at inference time to trade accuracy for latency without retraining (Figures 3 and 4), with grounding degrading earliest as throughput increases, whereas the block size B is fixed during training and offers a deployment-time choice among checkpoints. Orthogonal system-level optimizations, which change neither the model nor the decoding algorithm, compound the algorithmic gain: at $B = 32$ on NAVSIM, FlashInfer kernels (Ye et al., 2025b) for the block-causal mask reduce the per-token latency from 8.1 to 4.6 ms, and capturing each fixed-shape refinement step as a CUDA Graph (Ghosh et al., 2025) further reduces it to 1.5 ms, i.e., $15.1\times$ faster than the 22.7 ms of AR decoding (Figure 1).

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 raises PDMS from 87.7 to 88.3, indicating that the 2B student benefits from the stronger 8B teacher beyond what block-wise distillation provides.

Paradigm-Consistent Distillation Mechanics. We ablate both the teacher paradigm and the distillation objective. The 8B diffusion teacher shows higher top-K token overlap with the 2B diffusion student than the 8B AR teacher (Table 6): top-1 agreement is similar ($\rho_1 = 84.8\%$ vs. 83.6%), but a clear gap emerges at K=5 ($\rho_5 = 58.2\%$ vs. 51.2%). This weaker alignment is reflected in a large drop when distilling from the AR teacher (79.9 vs. 88.3 PDMS; Table 7), suggesting that paradigm consistency matters for effective cross-scale distillation. For the objective (Table 5), all

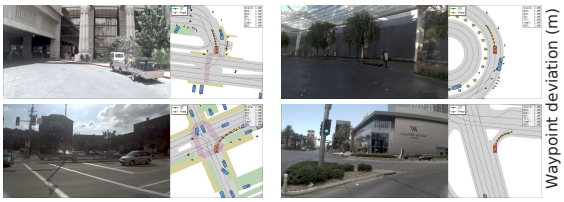


Figure 5: Qualitative results of WAM-Diff2 on the NAVSIM benchmark. The predicted trajectories demonstrate robust planning behavior across a range of diverse driving scenarios.

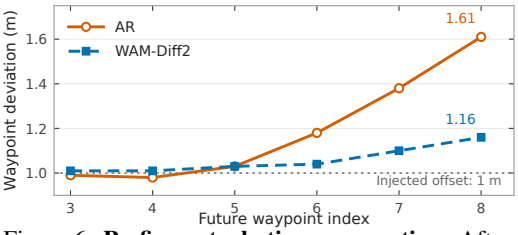


Figure 6: **Prefix-perturbation propagation.** After the first two ground-truth waypoints are shifted by 1 m. AR amplifies the offset, whereas WAM-Diff2 largely preserves it.

Table 8: **Robustness to controlled prefix perturbations.** The first two waypoints are set to ground truth (clean) or ground truth shifted by 1 m, and remaining waypoints are freely generated. Clean/perturbed FDE are measured against the ground truth, and ΔFDE is their difference. Final shift (mean), P90 shift, and P(Shift>1.5 m) measure displacement between perturbed and clean rollouts at waypoint 8. All distances are in meters.

Model	Clean FDE	Perturbed FDE	ΔFDE↓	Final Shift↓	P90 Shift↓	P(Shift>1.5 m)↓
AR	1.071	1.830	+0.759	1.61	3.30	36.8%
WAM-Diff2	0.998	1.456	**+0.458**	**1.16**	**1.85**	**19.2%**

three divergences perform comparably: JSD is marginally best (88.3 PDMS vs. 88.2 for forward KL and 88.0 for reverse KL) and is adopted as the default, but the small margin indicates that the choice of divergence is not a dominant factor.

Mitigation of Exposure Bias. WAM-Diff2 exhibits less error amplification under controlled prefix perturbations. On the NAVSIM test split, both models freely generate waypoints 3–8 from the same two-waypoint ground-truth prefix, either clean or shifted by 1 m. Displacement between perturbed and clean rollouts starts near the injected offset for both models but reaches 1.61 m for AR versus 1.16 m for WAM-Diff2 at waypoint 8 (Figure 6). FDE, measured against ground truth, increases less for WAM-Diff2 (0.46 vs. 0.76 m), while the fraction of final displacements exceeding 1.5 m falls from 36.8% to 19.2% (Table 8). Although blocks remain causally ordered, this reduction holds across waypoints 3–8, which span multiple blocks, suggesting that WAM-Diff2 mitigates exposure bias. The benefit is therefore expected mainly in open-loop evaluation, which scores all eight waypoints: Bench2Drive instead replans six waypoints from fresh observations at each step and mainly executes near-term predictions, on which the two models differ little (Appendix Figure 7), so no corresponding gain appears there.

Limitations and Failure Cases. Qualitative failure modes on the NAVSIM (Dauner et al., 2024) and Bench2Drive (Jia et al., 2024) benchmarks are visualized in Appendix Figures 11 and 12. 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 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 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× 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.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the language of the manuscript and for assisting with LaTeX formatting, such as citation commands and bibliography cleanup. We have not used generative AI tools for research ideation, method design, experiment implementation, data analysis, or the generation of experimental results. We have reviewed all AI-assisted work, and all AI-edited text and formatting changes were manually checked by the authors for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies efficient decoding for vision-language-action models in autonomous driving and does not involve human subjects or the collection of new data. All training and evaluation use publicly available datasets and simulators, namely NAVSIM, DriveLM, LingoQA, COCO, and Bench2Drive built on CARLA, which we use in accordance with their respective licenses. Our results are obtained exclusively on open-loop benchmarks and in closed-loop simulation; WAM-Diff2 has not been validated on real vehicles and should not be deployed in safety-critical settings without extensive real-world testing and independent safety assessment. In particular, parallel decoding trades some accuracy for speed (Section 4), and our failure cases (Appendix Figures 11 and 12) show that the model can still produce unsafe trajectories. Like other VLAs, WAM-Diff2 may also generate incorrect scene descriptions, so its language outputs should not be treated as reliable explanations of its driving decisions.

REPRODUCIBILITY STATEMENT

Section 3.1 defines the model formulation and the confidence-based decoding procedure, and Section 3.2 specifies the three training stages and their objectives. Section 4.1 describes the backbone, training data, optimizer settings, and evaluation protocols. The appendix provides the composition and size of every training dataset (Table 9), the definitions of all evaluation metrics (Eqs. (4)–(7)), stage-wise hyperparameters and compute budgets (Appendix A.3 and Table 10), and the exact prompt templates and output formats for all tasks (Appendix A.4). All models are initialized from publicly available Qwen3-VL checkpoints, and all datasets and benchmarks are public. We will release the code and model checkpoints upon acceptance.

REFERENCES

Amirhossein Afsharrad, Amirhesam Abedsoltan, Ahmadreza Moradipari, and Sanjay Lall. On-policy distillation of language models for autonomous vehicle motion planning. *arXiv preprint arXiv:2604.07944*, 2026.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Gregor Bachmann and Vaishnavh Nagarajan. The pitfalls of next-token prediction. *arXiv preprint arXiv:2403.06963*, 2024.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Zhepeng Cen, Yao Liu, Siliang Zeng, Pratik Chaudhari, Huzefa Rangwala, George Karypis, and Rasool Fakoor. Bridging the training-inference gap in llms by leveraging self-generated tokens. *arXiv preprint arXiv:2410.14655*, 2024.

Jian Chen, Yesheng Liang, and Zhijian Liu. DFlash: Block diffusion for flash speculative decoding. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026a.

Jiayi Chen, Wenxuan Song, Shuai Chen, Jingbo Wang, Zhijun Li, and Haoang Li. Dfm-vla: Iterative action refinement for robot manipulation via discrete flow matching. *arXiv preprint arXiv:2603.26320*, 2026b.

Yuntao Chen, Yuqi Wang, and Zhaoxiang Zhang. Drivinggpt: Unifying driving world modeling and planning with multi-modal autoregressive transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26890–26900, 2025.

Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, and Bowen Zhou. SDAR: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*, 2025.

Shuang Cheng, Yuhua Jiang, Zineng Zhou, Dawei Liu, Tao Wang, Linfeng Zhang, Biqing Qi, and Bowen Zhou. SDAR-VL: Stable and efficient block-wise diffusion for vision-language understanding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2026a.

Xin Cheng, Xingkai Yu, Chenze Shao, et al. DSpark: Confidence-scheduled speculative decoding with semi-autoregressive generation. *arXiv preprint arXiv:2607.05147*, 2026b.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving, 2022. URL https://arxiv.org/abs/2205.15997.

Can Cui, Yupeng Zhou, Juntong Peng, Sung-Yeon Park, Zichong Yang, Prashanth Sankaranarayanan, Jiaru Zhang, Ruqi Zhang, and Ziran Wang. Vilad: A large vision language diffusion framework for end-to-end autonomous driving. *arXiv preprint arXiv:2508.12603*, 2025.

Chenxu Dang, Sining Ang, Yongkang Li, Haochen Tian, Jie Wang, Guang Li, Hangjun Ye, Jie Ma, Long Chen, and Yan Wang. Drivefine: Refining-augmented masked diffusion vla for precise and robust driving. *arXiv preprint arXiv:2602.14577*, 2026.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

Erik Englesson and Hossein Azizpour. Generalized jensen-shannon divergence loss for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34:30284–30297, 2021.

Renju Feng, Ning Xi, Duanfeng Chu, Rukang Wang, Zejian Deng, Anzheng Wang, Liping Lu, Jinxiang Wang, and Yanjun Huang. Artemis: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving. *IEEE Robotics and Automation Letters*, 11(1):226–233, 2025.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. *arXiv preprint arXiv:2503.19755*, 2025.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

Abhishek Ghosh, Ajay Nayak, Ashish Panwar, and Arkaprava Basu. Pygraph: Robust compiler support for cuda graphs in pytorch. *arXiv preprint arXiv:2503.19779*, 2025.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2023.

Jianhua Han, Meng Tian, Jiangtong Zhu, Fan He, Huixin Zhang, Sitong Guo, Dechang Zhu, Hao Tang, Pei Xu, Yuze Guo, et al. Percept-wam: Perception-enhanced world-awareness-action model for robust end-to-end autonomous driving. *arXiv preprint arXiv:2511.19221*, 2025.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in neural information processing systems*, 34:12454–12465, 2021.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17853–17862, 2023.

Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, et al. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024.

Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. 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*, pp. 7953–7963, 2023a.

Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. 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*, pp. 21983–21994, 2023b.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 37:819–844, 2024.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8350, 2023.

Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv preprint arXiv:2410.22313*, 2024.

Jungho Kim, Jiyong Oh, Seunghoon Yu, Hongjae Shin, Donghyuk Kwak, and Jun Won Choi. Safedrive: Fine-grained safety reasoning for end-to-end driving in a sparse world. *arXiv preprint arXiv:2602.18887*, 2026.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Jingyu Li, Junjie Wu, Dongnan Hu, Xiangkai Huang, Bin Sun, Zhihui Hao, Xianpeng Lang, Xiatian Zhu, and Li Zhang. Sgdrive: Scene-to-goal hierarchical world cognition for autonomous driving. *arXiv preprint arXiv:2601.05640*, 2026a.

Pengxiang Li, Yinan Zheng, Yue Wang, Huimin Wang, Hang Zhao, Jingjing Liu, Xianyuan Zhan, Kun Zhan, and Xianpeng Lang. Discrete diffusion for reflective vision-language-action models in autonomous driving. *arXiv preprint arXiv:2509.20109*, 2025a.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35: 4328–4343, 2022.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026b.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, et al. Drivevla-w0: World models amplify data scaling law in autonomous driving. *arXiv preprint arXiv:2510.12796*, 2025b.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. *arXiv preprint arXiv:2506.08052*, 2025c.

Yongkang Li, Lijun Zhou, Sixu Yan, Bencheng Liao, Tianyi Yan, Kaixin Xiong, Long Chen, Hongwei Xie, Bing Wang, Guang Chen, et al. Unidrivevla: Unifying understanding, perception, and action planning for autonomous driving. *arXiv preprint arXiv:2604.02190*, 2026c.

Zekai Li, Yihao Liang, Hongfei Zhang, Jian Chen, Yesheng Liang, and Zhijian Liu. FlashDrive: Flash vision-language-action inference for autonomous driving. *arXiv preprint arXiv:2608.12932*, 2026d.

Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024b.

Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Tian Nian, Liuao Pei, Shunbo Zhou, Xiaokang Yang, Jiangmiao Pang, et al. Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint arXiv:2508.20072*, 2025.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12037–12047, 2025.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Yang Liu, Pengxiang Ding, Tengyue Jiang, Xudong Wang, Wenxuan Song, Minghui Lin, Han Zhao, Hongyin Zhang, Zifeng Zhuang, Wei Zhao, et al. Mmada-vla: Large diffusion vision-language-action model with unified multi-modal instruction and generation. *arXiv preprint arXiv:2603.25406*, 2026.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Yingzi Ma, Yulong Cao, Jiachen Sun, Marco Pavone, and Chaowei Xiao. Dolphins: Multimodal language model for driving. In *European Conference on Computer Vision*, pp. 403–420. Springer, 2024.

Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415*, 2023.

Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, et al. Lingoqa: Visual question answering for autonomous driving. In *European Conference on Computer Vision*, pp. 252–269. Springer, 2024.

Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving. In *European Conference on Computer Vision*, pp. 292–308. Springer, 2024.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11993–12003, 2025.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15120–15130, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pp. 256–274. Springer, 2024.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Dimitri Von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. *arXiv preprint arXiv:2503.04482*, 2025.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In *Proceedings of the computer vision and pattern recognition conference*, pp. 22442–22452, 2025a.

Tao Wang, Cong Zhang, Xingguang Qu, Kun Li, Weiwei Liu, and Chang Huang. Diffad: A unified diffusion modeling approach for autonomous driving. *arXiv preprint arXiv:2503.12170*, 2025b.

Junjie Wen, Minjie Zhu, Jiaming Liu, Zhiyuan Liu, Yicun Yang, Linfeng Zhang, Shanghang Zhang, Yichen Zhu, and Yi Xu. dvla: Diffusion vision-language-action model with multimodal chain-of-thought. *arXiv preprint arXiv:2509.25681*, 2025a.

Yuqing Wen, Hebei Li, Kefan Gu, Yucheng Zhao, Tiancai Wang, and Xiaoyan Sun. Llada-vla: Vision language diffusion action models. *arXiv preprint arXiv:2509.06932*, 2025b.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dLLM: Training-free acceleration of diffusion LLM by enabling KV cache and parallel decoding. *arXiv preprint arXiv:2505.22618*, 2025.

Chengyue Wu, Shiyi Lan, Yonggan Fu, Sensen Gao, Jin Wang, Jincheng Yu, Jose M. Alvarez, Pavlo Molchanov, Ping Luo, Song Han, Ligeng Zhu, and Enze Xie. Fast-dVLM: Efficient block-diffusion VLM via direct conversion from autoregressive VLM. *arXiv preprint arXiv:2604.06832*, 2026a.

Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. Fast-dLLM v2: Efficient block-diffusion LLM. In *International Conference on Learning Representations*, 2026b.

Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 35:6119–6132, 2022.

Shaoyuan Xie, Lingdong Kong, Yuhao Dong, Chonghao Sima, Wenwei Zhang, Qi Alfred Chen, Ziwei Liu, and Liang Pan. 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*, pp. 6585–6597, 2025.

Mingwang Xu, Jiahao Cui, Feipeng Cai, Hanlin Shang, Zhihao Zhu, Shan Luan, Yifang Xu, Neng Zhang, Yaoyi Li, Jia Cai, et al. Wam-diff: A masked diffusion vla framework with moe and online reinforcement learning for autonomous driving. *arXiv preprint arXiv:2512.11872*, 2025a.

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 9(10):8186–8193, 2024.

Zhenhua Xu, Yan Bai, Yujia Zhang, Zhuoling Li, Fei Xia, Kwan-Yee K Wong, Jianqiang Wang, and Hengshuang Zhao. Drivegpt4-v2: Harnessing large language model capabilities for enhanced closed-loop autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17261–17270, 2025b.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025.

Zhenjie Yang, Yilin Chai, Xiaosong Jia, Qifeng Li, Yuqian Shao, Xuekai Zhu, Haisheng Su, and Junchi Yan. 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*, pp. 10678–10688, 2026.

Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M Alvarez, and Zuxuan Wu. Drivesuprim: Towards precise trajectory selection for end-to-end planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11910–11918, 2026.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025a.

Zihao Ye, Lequn Chen, Ruihang Lai, Wuwei Lin, Yineng Zhang, Stephanie Wang, Tianqi Chen, Baris Kasikci, Vinod Grover, Arvind Krishnamurthy, et al. Flashinfer: Efficient and customizable attention engine for llm inference serving. *Proceedings of Machine Learning and Systems*, 7, 2025b.

Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*, 2025.

Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint arXiv:2305.10430*, 2023.

Jiaru Zhang, Manav Gagvani, Can Cui, Juntong Peng, Ruqi Zhang, and Ziran Wang. Efficient and explainable end-to-end autonomous driving via masked vision-language-action diffusion. *arXiv preprint arXiv:2602.20577*, 2026.

Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Li, Xianyuan Zhan, et al. Diffusion-based planning for autonomous driving with flexible guidance. In *International Conference on Learning Representations*, volume 2025, pp. 37207–37227, 2025.

Zhiyu Zheng, Shaoyu Chen, Haoran Yin, Xinbang Zhang, Jialv Zou, Xinggang Wang, Qian Zhang, and Lefei Zhang. Resad: Normalized residual trajectory modeling for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3729–3739, 2026.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, and Alois C Knoll. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. *arXiv preprint arXiv:2503.23463*, 2025a.

Zewei Zhou, Tianhui Cai, Seth Z Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *arXiv preprint arXiv:2506.13757*, 2025b.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025. URL https://arxiv.org/abs/2504.10479.

Jialv Zou, Shaoyu Chen, Bencheng Liao, Zhiyu Zheng, Yuehao Song, Lefei Zhang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Diffusiondrivev2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving, 2025. URL https://arxiv.org/abs/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 `navtrain` 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 (Lin et al., 2014): 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 9: Overview of the datasets used for training.

Dataset	**Domain**	**# Samples**	**Task**
		Unified Multi-Task Split	
NAVSIM	Driving	103,288	Planning
DriveLM	Driving	346,978	Driving VQA
LingoQA	Driving	413,829	Driving VQA
COCO-D	Vision	117,266	Detection
		Task-Specific Splits	
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):

$$\mathrm{PDMS} = \mathrm{NC} \cdot \mathrm{DAC} \cdot \frac{5\,\mathrm{EP} + 5\,\mathrm{TTC} + 2\,\mathrm{Comf}}{12}. \tag{4}$$

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):

$$\begin{aligned}\text{EPDMS} = \text{NC} \cdot \text{DAC} \cdot \text{DDC} \cdot \text{TLC} \\ \cdot \frac{5\,\text{EP} + 5\,\text{TTC} + 2\,\text{LK} + 2\,\text{HC} + 2\,\text{EC}}{16}.\end{aligned} \tag{5}$$

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:

$$\text{SR} = \frac{N_{\text{success}}}{N_{\text{total}}}, \tag{6}$$

$$\text{DS} = \frac{1}{N_{\text{total}}} \sum_{i=1}^{N_{\text{total}}} \left(\text{RC}_i \times \prod_{j=1}^{M_i} \text{IS}_{i,j} \right), \tag{7}$$

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 pre-trained Qwen3-VL backbones at two scales, using the official Qwen3-VL vision encoders that follow the SigLIP-2 architecture (Tschannen et al., 2025). The 8B variant pairs a SigLIP2-SO-400M encoder (27 blocks, 1152 hidden dimensions) with a 36-block language backbone, and the 2B variant pairs a SigLIP2-Large (300M) encoder (24 blocks, 1024 hidden dimensions) with a 28-block backbone. A dynamic-resolution strategy with 2D RoPE (Su et al., 2024) accommodates arbitrary camera aspect ratios. 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 and Block-Wise Distillation (Stages I and II). To manage the severe architectural shift from sequential to parallel generation, we incrementally relax causal constraints by scaling the decoding block size:

$$\text{AR} \to B_4 \to B_8 \to B_{16} \to B_{32}. \tag{8}$$

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}. For each larger block size ($B > 4$), the student is further distilled from the preceding block-$B/2$ diffusion model along the chain $B_4 \to B_8 \to B_{16} \to B_{32}$ (Stage II), optimizing a symmetric Jensen-Shannon

Table 10: 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

Divergence (JSD) loss over intermediate states sampled from the student's own decoding with a unified learning rate of 2×10^{-6} across all components.

Phase 3: Model-Wise Cross-Scale Distillation (Stage III). To close the capacity gap resulting from compact parameterization, the 2B block-32 student model is distilled for 5 epochs from an 8B block-32 diffusion teacher, optimizing the same JSD objective with 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 10. Closed-loop evaluations and all other model-quality evaluations are performed on a single node utilizing eight Ascend 910C NPUs. All decoding throughput (TPS) and latency measurements, including the Flash-Infer and CUDA Graph optimizations, 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 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.

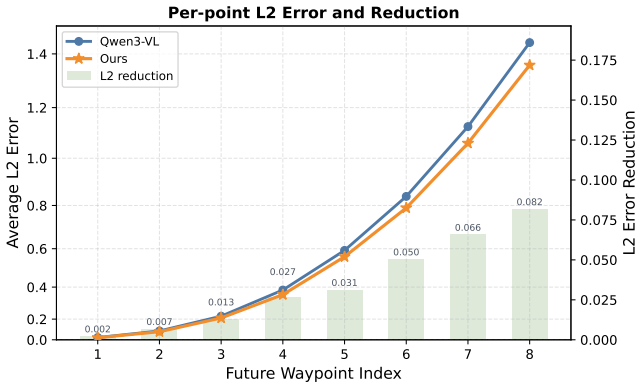


Figure 7: Per-waypoint L2 error under inference on the NAVSIM dataset. The L2 errors of the autoregressive baseline gradually increase compared to our method, and there is a trend of error amplification. These results suggest that bidirectional diffusion generation helps to alleviate error accumulation caused by autoregressive exposure bias.

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 Per-Waypoint Error Analysis

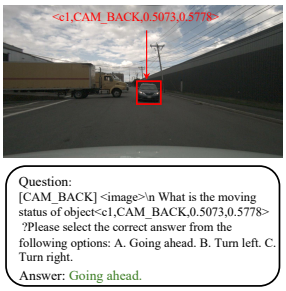
We compare the per-waypoint L_2 error of WAM-Diff2 and the autoregressive baseline across 12,146 paired NAVSIM samples (Figure 7). WAM-Diff2 reduces the average per-waypoint L_2 error by 5.8% (from 0.5935 to 0.5589), with the absolute reduction growing from 0.002 at the first waypoint to 0.082 at waypoint 8.

A.6 Additional Qualitative Results

We present additional qualitative results of WAM-Diff2, including closed-loop planning on Bench2Drive (Figure 8), driving understanding and visual perception (Figures 9 and 10), failure cases on NAVSIM and Bench2Drive (Figures 11 and 12), and representative Bench2Drive scenarios under diverse weather and lighting conditions (Figures 13 and 14).



Figure 8: Qualitative results of WAM-Diff2 on Bench2Drive. The predicted trajectories demonstrate robust planning behavior across diverse driving scenarios.



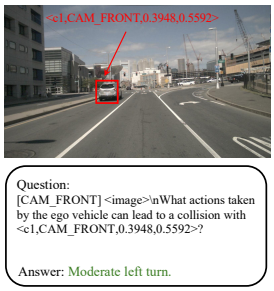


Figure 9: Visualization of understanding results.



Figure 10: Visualization of perception results.

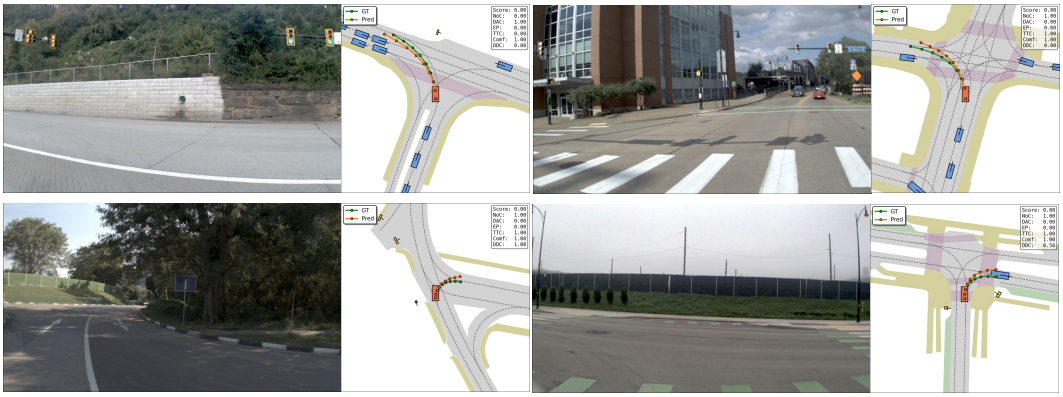


Figure 11: Failure cases visualization on NAVSIM.



Figure 12: Failure cases visualization on Bench2Drive.



Figure 13: Representative extreme-weather driving scenarios on Bench2Drive, specifically featuring Night & Fog, Dense Fog, and Rain & Fog conditions.



Figure 14: Representative driving scenarios in Bench2Drive under diverse weather and lighting conditions, including (a) HazardAtSideLane, (b) ParkingExit, and (c) VehicleOpensDoorTwoWays.