This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

WAM-Flow: Parallel Coarse-to-Fine Motion Planning via Discrete Flow Matching for Autonomous Driving

Yifang Xu[1*] Jiahao Cui[1*] Zhihao Zhu[1*] Hanlin Shang[1] Shan Luan[1] Mingwang Xu[1]
Feipeng Cai[2] Neng Zhang[2] Yaoyi Li[2] Jia Cai[2] Siyu Zhu[1✉]
[1]Fudan University [2]Yinwang Intelligent Technology Co., Ltd

Code & Model: https://github.com/fudan-generative-vision/WAM-Flow

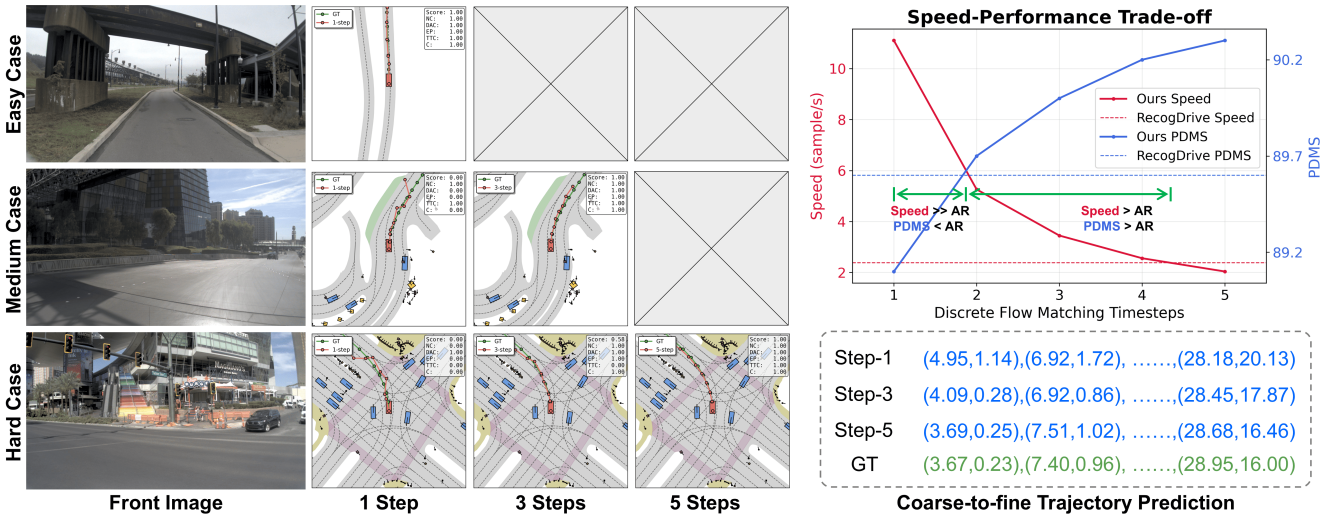


Figure 1. WAM-Flow enables flexible slow–fast and coarse-to-fine trajectory prediction. For straightforward driving scenarios, 1-step denoising achieves competitive performance (89.1 PDMS on NAVSIM-v1), while complex situations benefit from 5-step refinement, yielding further gains (90.3 PDMS). This corresponds to an inference speedup of 4.67× over RecogDrive [27] with 1-step denoising, while 5-step processing matches RecogDrive's latency. These results demonstrate the potential of discrete flow matching for building reliable and scalable autonomous driving systems.

Abstract

We introduce WAM-Flow, a vision–language–action (VLA) model that casts ego-trajectory planning as discrete flow matching over a structured token space. In contrast to autoregressive decoders, WAM-Flow performs fully parallel, bidirectional denoising, enabling coarse-to-fine refinement with a tunable compute–accuracy trade-off. Specifically, the approach combines a metric-aligned numerical tokenizer that preserves scalar geometry via triplet-margin learning, a geometry-aware flow objective and a simulator-guided GRPO alignment that integrates safety, ego progress, and comfort rewards while retaining parallel generation. A multi-stage adaptation converts a pre-trained auto-regressive backbone (Janus-1.5B) from causal decoding to non-causal flow model and strengthens road-scene competence through continued multimodal pretraining. Thanks to the inherent nature of consistency model training and parallel decoding inference, WAM-Flow achieves superior closed-loop performance against autoregressive and diffusion-based VLA baselines, with 1-step inference attaining 89.1 PDMS and 5-step inference reaching 90.3 PDMS on NAVSIM v1 benchmark. These results establish discrete flow matching as a new promising paradigm for end-to-end autonomous driving.

1. Introduction

Vision-language–action models for end-to-end autonomous driving [16, 27, 44, 50, 54] aim to map egocentric driving-

*: Equal contribution. {xuyf25, cuijh25}@m.fudan.edu.cn
✉: Corresponding authors. siyuzhu@fudan.edu.cn

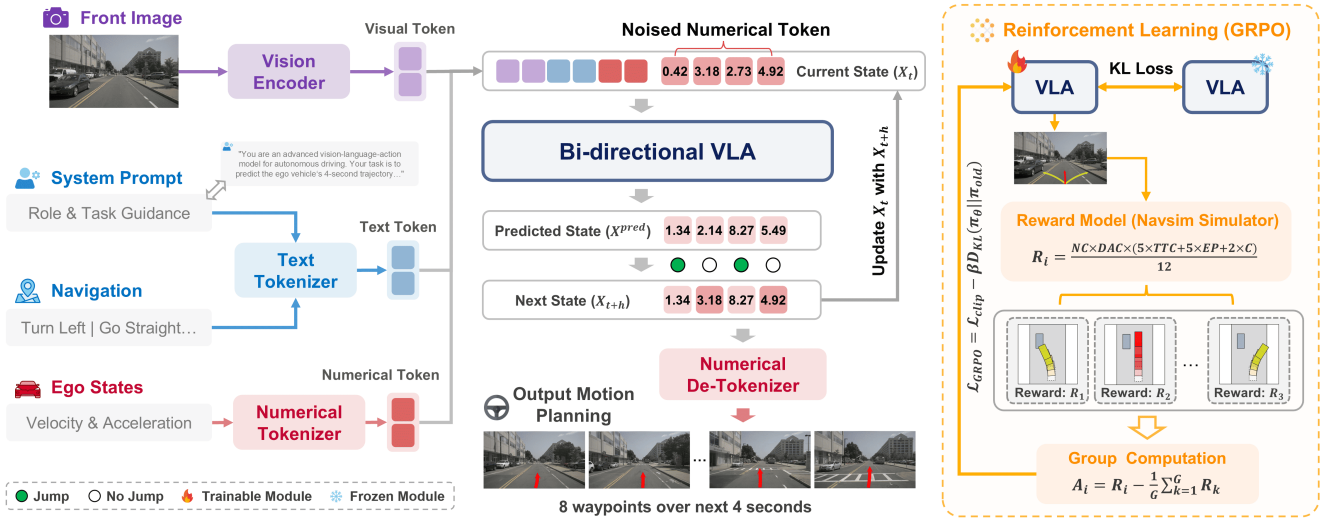


Figure 2. Architecture of the proposed WAM-Flow framework. Our method takes as input a front-view image, a natural-language navigation command with a system prompt, and the ego-vehicle states, and outputs an 8-waypoint future trajectory spanning 4 seconds through parallel denoising. The model is first trained via supervised fine-tuning to learn accurate trajectory prediction. We then apply simulator-guided GRPO to further optimize closed-loop behavior. The GRPO reward function integrates safety constraints (collision avoidance, drivable-area compliance) with performance objectives (ego-progress, time-to-collision, comfort).

view video inputs and natural-language instructions into both causal reasoning and precise ego-vehicle motion planning, while satisfying stringent efficiency and safety requirements. A fundamental challenge in this domain is the design of a policy representation that effectively balances three critical aspects: expressive reasoning capabilities, high-fidelity continuous control, and robust closed-loop performance. Existing approaches can be broadly categorized into dual-system and single-system paradigms. Dual-system methods [18, 27, 41, 52, 53] typically employ autoregressive vision-language models (VLMs) [2, 30, 38, 40, 55] as auxiliary reasoning modules to provide high-level driving intent, scene summaries, or linguistic guidance for downstream motion planning networks, which often utilize diffusion-based iterative optimization [11, 17, 20, 28] to generate smooth, complex action distributions. In contrast, single-system approaches [5, 19, 43, 44, 50, 54] such as EMMA [19] and DrivingGPT [5] reformulate trajectory or action prediction as a text generation problem within the VLM, enabling reasoning and planning directly in the linguistic space. This work investigates a novel alternative based on discrete flow matching (DFM), which offers distinct advantages for autonomous driving applications.

Discrete flow matching [6, 10, 12, 23, 35, 36, 39, 49] models probability transport over discrete token spaces via a continuous-time Markov chain (CTMC) that carries a simple base distribution to the data distribution. Unlike autoregressive decoders that commit to tokens sequentially and accumulate exposure-bias errors, Discrete flow matching supports fully parallel denoising and bidirectional refinement during generation. These properties enable coarse-to-fine planning: beginning with a coarse motion hypothesis, the model increases trajectory fidelity through additional denoising steps, yielding a tunable compute–accuracy trade-off. This flexibility aligns well with autonomous driving, where simple scenes admit rapid approximate plans while complex interactions require higher-precision refinement. Despite these advantages, discrete flow matching remains largely unexplored for VLA policies in end-to-end autonomous driving.

However, a straightforward application of discrete flow matching to VLA model for end-to-end autonomous driving is nontrivial for three reasons. First, training discrete flow matching from scratch is prohibitively data- and compute-intensive, so they are typically initialized from general-purpose autoregressive multimodal VLMs that lack sufficient road-scene competence–from low-level perception and motion forecasting to high-level planning and decision making. We therefore adopt a multi-stage adaptation strategy: starting from a generic VLM backbone (Janus-1.5B [42]), we continued conduct pretraining on large-scale road-scene visual question answering (VQA) to strengthen the ability to understand various complex road scenes and vehicle driving patterns, establishing a strong domain prior comparable to autoregressive VLA baselines. Second, standard text token embeddings are ill-suited to high-precision numerical regression because they weakly encode metric relationships. We introduce a metric-aligned numerical tokenizer that discretizes continuous scalars into a shared codebook and learns embeddings with a triplet-

margin ranking objective so that latent distances reflect underlying scalar differences. This structured token space enables stable coarse-to-fine and slow–fast trajectory refinement within discrete flow matching, providing a controllable compute–accuracy trade-off. Finally, supervised likelihood-based flow training aligns the model with expert trajectories but does not explicitly enforce safety, ego-progress, and comfort in closed-loop control. We incorporate a Group Relative Policy Optimization (GRPO) based alignment objective with a composite reward that integrates safety penalties and performance goals, improving the safety–progress–comfort profile while preserving the model's parallel generation capabilities.

Experimental results on the NAVSIM v1 and v2 benchmarks demonstrate that WAM-Flow achieves superior performance in PDMS and EPDMS metrics compared to both autoregressive and diffusion-based VLA models. By leveraging discrete flows over a structured token space, WAM-Flow enables flexible slow–fast and coarse-to-fine trajectory prediction. With 1-step denoising, it attains competitive performance (89.1 PDMS), while 5-step refinement yields further gains (90.3 PDMS). On the NAVSIM v2 benchmark, the full model achieves 84.7 EPDMS. Notably, the 1.5B-parameter WAM-Flow model achieves an $3\times$ improvement in inference speed over the Janus autoregressive baseline, underscoring the promising effectiveness and efficiency of the discrete flow matching approach for end-to-end autonomous driving.

2. Related Work

VLMs in Autonomous Driving. Autoregressive VLMs [19, 21, 37, 43, 50, 54] formulate driving as a sequential language modeling problem, where each token corresponds to a trajectory point, control command, or reasoning step. Representative works such as EMMA [19], OpenEMMA [43], FutureSightDrive [50] and AutoVLA [54] leverage chain-of-thought reasoning and external memory modules to enhance interpretability and decision transparency. Despite their strong causal modeling capability, autoregressive architectures suffer from slow autoregressive decoding and limited parallelism, as future actions must be generated step-by-step. Diffusion-based methods [8, 20, 28, 44], including DiffusionDrive [28], ViLaD [8] and WAM-Diff [44] treat planning as a denoising process that gradually refines latent trajectory representations. These models enable parallel sampling but often lack explicit reasoning interpretability. In this paper, we explore a new promising paradigm for end-to-end autonomous driving, namely discrete flow matching.

Discrete Diffusion in LLMs and VLMs. Recent progress in discrete generative modeling has led to the emergence of discrete diffusion LLMs [32, 46] and discrete diffusion VLMs [26, 31, 45, 47, 48], which extend diffusion processes to tokenized sequences. This direction originates from D3PM [1], which formulated diffusion as a discrete Markov process over categorical variables. Recently, LLaDA [32] trains an 8B-parameter model from scratch, reaching LLaMA-3 [14] performance with bidirectional reasoning and robustness. DREAM-7B [46] further enhances diffusion-based reasoning via iterative refinement and arbitrary-order generation. Meanwhile, discrete flow matching [13, 29, 34] generalizes discrete diffusion via continuous-time probability paths and learnable velocity fields. By unifying diffusion and flow-based generation under a single probabilistic framework, it enables parallel, bidirectional, and efficient sampling. FUDOKI [39] extends this framework to multimodal reasoning and generation, demonstrating unified, non-autoregressive modeling across modalities. In this paper, we apply discrete flow matching to VLA for autonomous driving and explore its inherent nature of parallel generation and coarse-to-fine controllability.

Reinforcement Learning in VLA. Building upon the success of DeepSeek-R1 [15], GRPO has been further extended to autonomous driving domains. In particular, AlphaDrive [22] pioneers the integration of GRPO-based reinforcement learning with planning-centric reasoning in autonomous driving, achieving notable improvements in both decision-making performance and training efficiency. TrajHF [24] further combines diffusion-based multimodal planners with reinforcement learning from human feedback, enabling safe and personalized trajectory generation aligned with diverse human driving styles. More recently, AutoVLA [54] incorporates GRPO into vision-language-action models, extending reinforcement learning to end-to-end multimodal reasoning and low-level planning. To the best of our knowledge, this work presents the first exploration of GRPO within discrete flow matching for autonomous driving VLA. Furthermore, we explicitly incorporate safety alignment objectives, extending beyond conventional likelihood-based training to enhance reliability in autonomous driving contexts.

3. Method

We present WAM-Flow, a VLA model that formulates motion planning as a discrete flow matching problem over a structured token space. Specifically, Section 3.1 establishes the theoretical foundation of discrete flow matching over finite alphabets. Building on this, Section 3.2 details the model architecture, including a metric-aligned numerical tokenizer, and a geometry-aware flow objective. To address the limitations of likelihood-based training, Section 3.3 introduces simulator-guided GRPO to enforce safety and performance in closed-loop control. Finally, Section 3.4 specifies the autoregressive-to-flow training and the parallel denoising–based inference. Figure 2 demonstrates the pipeline of WAM-Flow.

3.1. Preliminaries: Discrete Flow Matching

Probability Paths. Let the discrete state space be defined as $S = \mathcal{T}^D$, where $\mathcal{T} = [K] = \{1, \dots, K\}$ represents a set of possible discrete values, and D is the number of discrete variables. Denote the data distribution by $q(x)$ over S and a simple factorized source distribution by $p(x) = \prod_{i=1}^{D} p^i(x^i)$. We define a time-dependent probability path $\{p_t(x)\}_{t\in[0,1]}$ by marginalizing conditional, coordinate-wise factorized paths around a latent target x_1:

$$p_t(x) = \sum_{x_1 \in S} q(x_1)\, p_t(x|x_1),\ p_t(x|x_1) = \prod_{i=1}^{D} p_t^i\big(x^i|x_1^i\big), \quad (1)$$

with boundary conditions ensuring $p_0^i(\cdot|x_1^i) = p^i(\cdot)$ and $p_1^i(\cdot|x_1^i) = \delta_{x_1^i}(\cdot)$, which yields $p_0(x) = p(x)$ and $p_1(x) = q(x)$. This mixture construction separates the definition of the transport path from the generative dynamics. A common instance is the mixture (mask) path:

$$p_t^i(x^i|x_1^i) = \big(1 - \kappa_t\big)\, p^i(x^i) + \kappa_t\, \delta_{x_1^i}(x^i), \quad (2)$$

where $\kappa_t \in [0, 1]$ is a monotonically increasing scheduling function satisfying $\kappa_0 = 0$ and $\kappa_1 = 1$. When $p^i(x^i) = \delta_{[\text{MASK}]}(x^i)$, this path recovers the standard masked corruption process.

Generative Dynamics. The probability path $p_t(x)$ is realized through a CTMC characterized by a probability velocity $u_t(x, z)$. This velocity acts as a rate matrix, defining the instantaneous transition rate from state z to state x at time t. Formally, for a small time step $h > 0$, the transition probability satisfies:

$$P(x_{t+h} = x|x_t = z) = \delta_z(x) + hu_t(x, z) + o(h), \quad (3)$$

where $\delta_z(x)$ is the Kronecker delta and $o(h)$ denotes higher-order terms. The velocity u_t must adhere to the constraints: $u_t(x, z) \geq 0$ for all $x \neq z$, and $\sum_x u_t(x, z) = 0$. This velocity generates the path p_t via the Kolmogorov forward equation:

$$\dot{p}_t(x) + \text{div}_x(j_t) = 0, \quad (4)$$

where the probability flux is given by $j_t(x, z) = u_t(x, z)p_t(z)$. To maintain tractability in high-dimensional spaces, we restrict the velocity to permit only single-coordinate transitions.

3.2. WAM-Flow Architecture

Problem Formulation. We formulate the motion planning task as a conditional sequence generation problem. The model maps multimodal inputs–including synchronized front-view camera images, a natural-language navigation command, and the current ego-vehicle state (position, heading, velocity and acceleration)–to a discrete token sequence representing the planned trajectory. The output is a sequence of 8 waypoints spanning the next 4 seconds.

Within this formulation, WAM-Flow employs a flow network that learns to transport a simple prior distribution over the discrete token space to the expert trajectory distribution. An advantage of this approach is its support for fully parallel token transitions during generation, which circumvents the sequential bottleneck of autoregressive decoding. This capability enables a flexible trade-off between computational efficiency and prediction fidelity: rapid, coarse plans can be generated with few denoising steps, while high-precision trajectories are achieved through iterative refinement.

Metric-Aligned Numerical Tokenizer. Standard text token embeddings do not preserve metric structure and thus perform poorly for high-precision regression. We introduce a metric-aligned numerical tokenizer that discretizes continuous scalars (e.g., position, heading, velocity and acceleration) into a uniform codebook $\mathcal{V} = \{v_1, \dots, v_N\}$ over $[-100, 100]$ with 0.01 resolution ($N = 20{,}001$). Each scalar token v is mapped by a linear projection $E : \mathbb{R} \to \mathbb{R}^d$ and L2-normalized to yield the embedding $z = E(v)/\|E(v)\|_2$.

To align latent geometry with numeric distances, we enforce that Euclidean embedding distances are monotonic in the underlying scalar differences. Let $d_{ij} = \|z_i - z_j\|_2$. For any triplet (i, j, k) with $|v_i - v_j| < |v_i - v_k|$, we promote $d_{ij} < d_{ik}$ via a triplet-margin ranking loss:

$$\mathcal{L}_{\text{num}} = \mathbb{E}_{(i,j,k)\sim\mathcal{T}}\big[\max\big(0,\ d_{ij} - d_{ik} + \alpha\big)\big], \quad (5)$$

where $\mathcal{T}$ samples anchors i with near/far neighbors (j, k) and $\alpha > 0$ is a fixed margin. This construction yields a numerically coherent token space in which latent distances faithfully reflect scalar proximity, enabling stable coarse-to-fine and slow–fast refinement under discrete flow matching. The induced distances serve as the tokenizer-specific metric $d_i(\cdot, \cdot)$ in the geometry-aware flow objective.

Discrete Flow Matching Objective. Consider that the geometric structure of the tokenized action space, we design a conditional probability path that is both tractable and expressive. Given a target sequence $x_1 \in q(x)$, we define a Gibbs distribution induced by a distance metric d:

$$p_t(x|x_1) = \text{softmax}\left(-\beta_t d(x, x_1)\right),\ \beta_0 = 0,\ \beta_1 \to \infty, \quad (6)$$

where β_t is a monotonically increasing scheduling function on $[0, 1]$, and $d(x, x_1) = \sum_{i=1}^{D} w_i d_i(x^i, x_1^i)$ is a weighted sum of coordinate-wise dissimilarities. Each d_i is tailored to the data type: tokenizer-induced distances for numerical values, circular metrics for angles, and semantic distances for textual fields. The nonnegative weights w_i balance the contribution of each coordinate.

This path is realized by a CTMC with a transition rate designed to steer the state toward the target. The conditional rate for transitioning from z to x given x_1 is:

$$u_t(x, z|x_1) = p_t(x|x_1)\dot{\beta}_t\left[d(z, x_1) - d(x, x_1)\right]_+, \quad (7)$$

where $[\cdot]_+ = \max(0, \cdot)$. This rate assigns higher probability to transitions that reduce the dissimilarity to the target. The marginal velocity is obtained by integrating over the posterior distribution of x_1 given the current state.

The model is trained to approximate the true posterior $p_{1|t}(x_1|x)$ by minimizing the conditional flow matching cross-entropy loss:

$$\mathcal{L}_{\text{CE}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, x_1\sim q,\, x\sim p_t(\cdot|x_1)} \left[-\sum_{i=1}^{D} \log p_{1|t}^{\theta,i}(x_1^i|x) \right], \tag{8}$$

where $p_{1|t}^{\theta,i}(x_1^i|x)$ is the model's estimate of the posterior probability for the i-th target token. This geometry-aware formulation enables efficient parallel decoding and supports controllable refinement, allowing flexible trade-offs between planning speed and trajectory quality.

Model Architecture. We adapt a Janus-1.5B multimodal backbone to the discrete flow matching generation paradigm for vision–language–action planning. Images are resized with preserved aspect ratio, zero-padded to 384×384, and encoded by SigLIP [51] into 576 visual tokens; a lightweight MLP aligns these features to the 2048-dimensional Janus text-token space. On the language side, we extend the Janus tokenizer by 20,001 numerically grounded tokens to represent input ego-state numbers and output waypoint coordinates, yielding a 122,401-word vocabulary. Training data are formatted with a fixed QA-style prompt that integrates navigation commands, ego-state (position, heading, velocity, acceleration), and the target waypoint sequence for the next 4 seconds (8 waypoints). For the decoder, the original Janus text head is expanded to the enlarged vocabulary and used to predict action tokens under the discrete flow matching objective.

3.3. Simulator-Guided GRPO

While supervised flow matching optimizes trajectory prediction accuracy, it does not explicitly enforce critical driving objectives such as safety, comfort, and progress in closed-loop control. To address this limitation, we introduce an online GRPO reinforcement learning that aligns the policy with simulator-derived rewards while preserving the parallel generation capabilities of discrete flow matching.

Reward Design. We design a composite reward function that decomposes the NAVSIM simulator's PDMS metrics into safety penalties and performance objectives. The reward for a generated trajectory τ is defined as:

$$R(\tau) = \underbrace{\left(\prod_{m\in\mathcal{M}} s_m(\tau) \right)}_{\text{safety penalties}} \cdot \underbrace{\left(\frac{\sum_{w\in\mathcal{W}} \lambda_w s_w(\tau)}{\sum_{w\in\mathcal{W}} \lambda_w} \right)}_{\text{performance objectives}}, \tag{9}$$

where $\mathcal{M} = \{\text{NC}, \text{DAC}\}$ represents safety metrics, including no-collision and drivable-area compliance; $\mathcal{W} = \{\text{EP}, \text{TTC}, \text{C}\}$ denotes performance metrics, including ego-progress, time-to-collision, and comfort. The multiplicative safety term ensures strict constraint satisfaction, while the weighted average balances performance trade-offs. Specifically, the NC score assigns $s_{\text{NC}}(\tau) = 0$ for at-fault collisions, 0.5 for collisions with static objects, and 1 otherwise; DAC yields $s_{\text{DAC}}(\tau) = 0$ on violations and 1 otherwise. Sub-scores $s_w(\tau) \in [0, 1]$ are normalized, with $\lambda_w \geq 0$ as weighting coefficients.

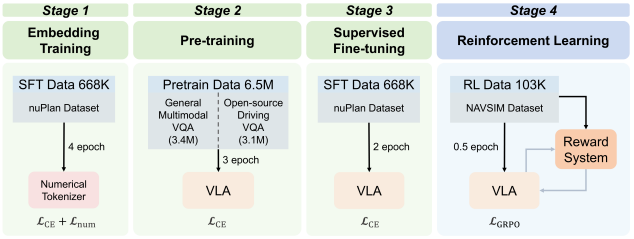


Figure 3. Overview of the full training curriculum. Different training stage motivation and corresponding training data and training steps are demonstrated.

GRPO Objective. For a given scene context c, we sample G candidate trajectories $\{\tau_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot|c)$ via parallel denoising. Each trajectory receives a reward $R_i = R(\tau_i)$. Using the group baseline $A_i = R_i - \frac{1}{G}\sum_{j=1}^{G} R_j$, we define per-token importance ratios for action tokens $\{o_i^k\}_{k=1}^{T_i}$ under conditioning states $\{s_i^k\}$: $r_i^k(\theta) = \frac{\pi_\theta(o_i^k s_i^k)}{\pi_{\theta_{\text{old}}}(o_i^k s_i^k)}$. The GRPO surrogate objective, with clipping parameter $\epsilon > 0$ and KL regularization strength $\beta \geq 0$, is formulated as:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_c \Bigg[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{T_i} \sum_{k=1}^{T_i} \Big(\min\Big\{ r_i^k(\theta)A_i, \text{clip}(r_i^k(\theta), 1-\epsilon, 1+\epsilon)A_i \Big\} - \beta D_{\text{KL}}\left(\pi_\theta(\cdot s_i^k)\, \pi_{\text{ref}}(\cdot s_i^k)\right) \Big) \Bigg]. \tag{10}$$

The group baseline reduces variance by inducing relative preferences within each sample set, while the KL divergence term stabilizes updates by anchoring the policy to the supervised reference.

3.4. Training and Inference

Autoregressive-to-Flow Training. Figure 3 outlines a four-stage curriculum. First, we randomly initialize the numerical embeddings and freeze the VLA backbone. We train the numerical embeddings together with the language-model head on the 668K nuPlan dataset for 4 epochs, using flow-matching loss $\mathcal{L}_{\text{CE}}$ (Equation 8) and triplet-margin ranking loss $\mathcal{L}_{\text{num}}$ (Equation 5). Second, we enhance the perception of driving scenes by pretraining VLA using $\mathcal{L}_{\text{CE}}$. This stage trains 3 epochs on 6.5M VQA, including general multimodal VQA (3.4M) from LLaVA-v1.5 [30] and large-scale driving-specific VQA (3.1M) from RecogDrive [27], which enhances perceptual grounding and

driving-specific causal reasoning. Third, we supervised fine-tuning of the VLA backbone for only 2 epochs on the nuPlan dataset with $\mathcal{L}_{\text{CE}}$. After supervised flow training, we perform reinforcement learning with simulator feedback by maximizing the GRPO objective (Equation 10) with KL regularization toward the supervised reference to optimize our VLA model for 0.5 epoch on 103k NAVSIM dataset. We set the weight for EP, TTC and C in our reward to 5:5:2.

Inference. First, we apply the Euler discretization over the time interval $[0, 1]$ with n inference steps, yielding a step size of $h = \frac{1}{n}$. For each coordinate i, the initial token x_0^i is sampled uniformly from the model vocabulary. At each discrete timestep $t \in [0, 1]$, the current token is denoted as x_t^i, and a target token x_1^i is drawn from the posterior distribution $p_{1|t}^i(x_1^i|x)$. Next, we compute the total outgoing transition rate λ_i for the current token x_t^i as $\lambda_i = \sum_{x^i \neq x_t^i} u_t^i(x^i, x_t^i|x_1^i)$, where the conditional rate function u_t^i is defined in Equation 7. A uniform random variable $Z_i \sim \mathcal{U}[0, 1]$ is then drawn. The jump rule is as follows: if $Z_i < 1 - e^{-h\lambda_i}$, a transition occurs, and the new token x_{t+h}^i is sampled from $\frac{u_t^i(\cdot, x_t^i|x_1^i)}{\lambda_i}\left(1 - \delta_{x_t^i}(\cdot)\right)$; otherwise, the token remains unchanged, i.e., $x_{t+h}^i = x_t^i$. After n sampling steps, we obtain the final output token sequence x_1.

4. Experiments

4.1. Experimental Setup

Implementation. All experiments were conducted on 4×8 Ascend 910B NPUs across four sequential training phases. We use AdamW optimizer for all training stages with weight decay of 0.01. In the metric-aligned numerical embeddings training stage, we set α to 0.05, constant learning rate to 1×10^{-5} and batch size to 80. In the pre-training stage, we set constant learning rate to 1×10^{-5} and batch size to 256. In the SFT stage, we utilize learning rate of 5×10^{-6} with cosine annealing strategy and batch size of 64. In the reinforcement learning stage, we use a learning rate of 1×10^{-6}, batch size of 32 and 500 warm-up steps. During inference, we use a timestep schedule defined as $\beta_t = 3 \times \left(\frac{t}{1-t}\right)^{0.9}$, and perform inference with 1, 2, 3, 5, and 10 sampling steps. On NAVSIM-v1 benchmark, we conduct evaluations using the v1.1 version of the NAVSIM codebase, while on NAVSIM-v2 benchmark, we use the v2.2 version for evaluation.

Metrics. We evaluate our method on the closed-loop NAVSIM-v1 [9] and v2 [3] benchmarks. The primary metric for NAVSIM-v1 is the Predictive Driver Model Score (PDMS), a composite measure integrating five key components: No-Collision rate (NC), Drivable Area Compliance (DAC), Time-to-Collision within bound (TTC), Comfort, and Ego Progress (EP). The more comprehensive NAVSIM-v2 benchmark employs the Extended Predictive Driver Model Score (EPDMS), which incorporates nine sub-metrics—NC, DAC, Driving Direction Compliance (DDC), Traffic Light Compliance (TLC), EP, TTC, Lane Keeping (LK), History Comfort (HC), and Extended Comfort (EC)– to provide a holistic assessment of driving performance, safety, and rule adherence. All results are obtained from closed-loop simulations on the official public test splits.

4.2. Comparison with State-of-the-Art

NAVSIM-v1. As shown in Table 1, our method achieves the highest PDMS (90.3) on the NAVSIM v1 benchmark. It attains the superior performance in both safety-critical metrics–No-Collision (NC: 99.2) and Drivable Area Compliance (DAC: 98.3)–demonstrating superior safety and rule adherence. Notably, despite utilizing only a single front-view camera, our model outperforms methods that rely on multi-view camera setups or LiDAR inputs, underscoring the efficacy of the discrete flow matching paradigm in achieving robust and efficient planning. Qualitative analyses (Figure 4 and Figure 5) further illustrate that our planner produces stable, human-like trajectories in closed-loop simulation.

NAVSIM-v2. Table 4 presents the evaluation results on the more comprehensive NAVSIM-v2 benchmark. Our method achieves the superior overall EPDMS of 84.7. It also leads in several critical sub-metrics: No-Collision (NC: 98.5), Driving Direction Compliance (DDC: 99.5), and Lane Keeping (LK: 97.4). The superior performance across diverse and dynamic scenarios underscores the robustness of our approach for reliable closed-loop driving.

4.3. Ablation and Discussion

Ablation Study for Proposed Components. As shown in Table 5, we systematically evaluate the contribution of each component in our framework. Using the Janus-1.5B text tokenizer for numerical values results in a PDMS of 76.2, indicating its inadequacy for representing fine-grained trajectory data. Replacing it with a dedicated numerical tokenizer improves PDMS by 4.9 points (to 81.1), confirming the necessity of a specialized numerical representation. Further incorporating metric-aligned embeddings yields an additional gain of 2.3 points (to 83.4), demonstrating that geometric consistency in the token space enhances planning quality. Subsequent large-scale VQA pretraining adds 3.3 points (to 86.7), underscoring the benefit of cross-modal domain adaptation. Finally, integrating simulator-guided GRPO achieves the highest PDMS of 90.3, highlighting the critical role of safety and performance alignment through online reinforcement learning.

In addition, comparing Rows 5 and 6 in Table 5 reveals

Method	Paradigm	Backbone	Input	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
End-to-End									
VADv2 [4]	-	-	6×Cam	97.2	89.1	91.6	**100**	76.0	80.9
Transfuser [7]	-	-	3×Cam + L	97.7	92.8	92.8	**100**	79.2	84.0
Hydra-MDP++ [25]	-	-	3×Cam + L	98.3	96.0	94.6	**100**	78.7	86.5
Artemis [11]	Diff.	-	6×Cam	98.3	95.1	94.3	99.8	81.4	87.0
DiffusionDrive [28]	Diff.	-	3×Cam + L	98.2	96.0	94.8	**100**	82.2	88.1
End-to-End VLA									
DrivingGPT [5]	AR	LLaMA2-7B [38]	1×Cam	98.1	90.7	94.9	95.6	79.7	82.4
FSDrive [50]	AR	Qwen2-VL-2B [40]	6×Cam	98.2	93.8	93.3	99.9	80.1	85.1
Epona [52]	AR + Diff.	DiT-2.5B [33]	1×Cam	97.9	95.1	93.8	99.9	80.4	86.2
AutoVLA [54]	AR	Qwen2.5-3B [2]	3×Cam	98.4	95.6	**98.0**	99.9	81.9	89.1
ReCogDrive [27]	AR + Diff.	InternVL3-8B [55]	3×Cam	98.2	97.5	95.2	99.9	**83.5**	89.6
Ours	DFM	Janus-1.5B [42]	1×Cam	**99.2**	**98.3**	97.0	99.7	82.3	**90.3**

Table 1. Comparison on NAVSIM-v1 with closed-loop metrics. Abbreviation: Diff.(Diffusion), Comf.(Comfort), Cam (Camera), L (LiDAR).

Group Size	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
w/o GRPO	98.5	95.1	94.4	99.5	81.8	86.7
2	**99.4**	97.3	96.8	99.7	80.7	89.2
3	99.2	**98.3**	**97.0**	99.7	**82.3**	**90.3**
4	99.3	97.6	96.5	**99.8**	82.0	89.6

Table 2. Ablation on GRPO group size.

EP : TTC : C	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
5:20:2	**99.5**	**98.3**	**97.9**	99.6	80.1	89.7
5:5:8	99.4	98.1	96.9	**99.7**	82.1	90.1
20:5:2	99.4	98.1	96.4	99.3	**82.7**	90.0
5:5:2	99.2	**98.3**	**97.0**	**99.7**	82.3	**90.3**

Table 3. Ablation on different weight of Simulator-Guided reward. The default weight is 5:5:2 for Navsim simulator, and we adjust the scale of each weight by $4\times$ to obtain the new weight.

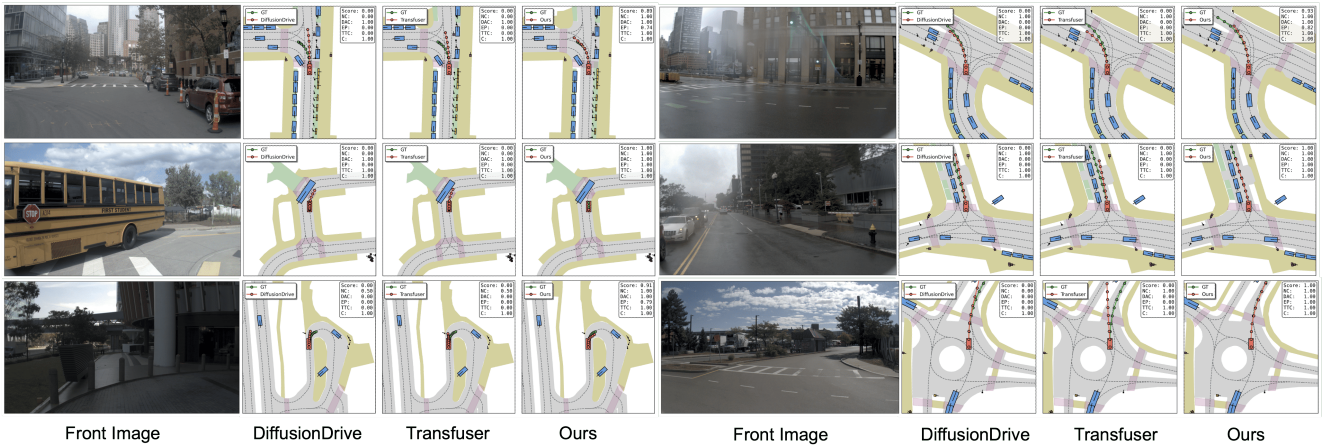


Figure 4. Qualitative comparison on NAVSIM.



Figure 5. Qualitative results of WAM-Flow on NAVSIM with different scenes.

that incorporating VQA pre-training yields a +3.4 improvement in PDMS, underscoring the efficacy of pre-training in enhancing driving performance. This gain demonstrates that domain-specific pre-training on large-scale visual question answering data provides valuable foundational knowledge for complex driving scenarios, complementing the benefits of reinforcement learning-based fine-tuning.

Further Pretrain on More Driving Data. Figure 6 illustrates the impact of pre-training epochs on model performance. For a fair comparison, we conduct SFT following pre-training on 6.5M VQA dataset. The results show that the PDMS score increases with the number of pre-training epochs, reaching a peak at 3 epochs, with an improvement of +3.3 compared to 0 epochs. This indicates

Method	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS↑
Ego Status	93.1	77.9	92.7	99.6	86.0	91.5	89.4	**98.3**	85.4	64.0
VADv2 [4]	97.3	91.7	98.2	99.7	77.6	92.7	66.0	**98.3**	83.3	76.6
TransFuser [7]	97.7	92.8	98.3	99.7	79.2	92.8	67.6	**98.3**	**87.2**	77.8
HydraMDP++ [25]	97.2	**97.5**	99.4	99.6	83.1	96.5	94.4	98.2	70.9	81.4
Artemis [11]	98.3	95.1	98.6	**99.8**	81.5	**97.4**	96.5	**98.3**	-	83.1
RecogDrive [27]	98.3	94.2	98.8	**99.8**	86.5	97.3	96.8	**98.3**	87.7	83.6
Ours	**98.5**	94.5	**99.5**	**99.8**	**86.9**	96.8	**97.4**	97.6	73.9	**84.7**

Table 4. Comparison on NAVSIM-v2 with extended metrics.

Numerical Tokenizer	Metric -aligned	Pre-training	SG GRPO	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
✗	✗	✗	✗	95.8	87.5	88.6	99.5	71.7	76.2
✓	✗	✗	✗	97.0	91.3	91.0	98.9	76.4	81.1
✓	✓	✗	✗	97.4	92.6	95.3	99.3	77.5	83.4
✓	✓	✓	✗	98.5	95.1	94.4	99.5	81.8	86.7
✓	✓	✗	✓	98.4	96.1	95.3	99.5	79.3	86.9
✓	✓	✓	✓	**99.2**	**98.3**	**97.0**	**99.7**	**82.3**	**90.3**

Table 5. Ablation study for the proposed components. We evaluate the effect of metric-aligned numerical tokenizer, VQA pretraining and simulator-guided GRPO on NAVSIM-v1. Row 1 uses the text tokenizer from Janus-1.5B to tokenize the number. "SG GRPO" refers to "Simulator-Guided GRPO".

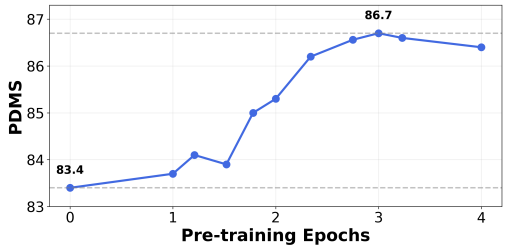


Figure 6. Impact of pre-training epochs. We perform SFT after pre-training on 6.5M data, and then calculate PDMS.

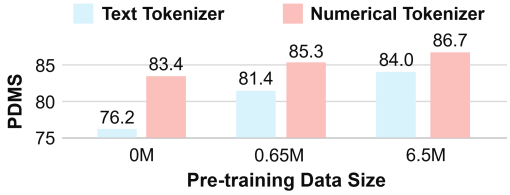


Figure 7. Effect of pretraining dataset scale.

that pre-training on driving-related VQA tasks significantly enhances the model's driving capabilities in road scenarios.

Figure 7 investigates the scaling laws of pre-training data. We found that pre-training with 0.65M data yielded a PDMS improvement of +1.9 (+5.2) for numerical (text) tokenizer. Further pre-training with 6.5M data resulted in an additional PDMS increase of +1.4 (+2.6) for numerical (text) tokenizer compared to the 0.65M data. These results highlight the necessity of further pre-training using driving VQA data and confirm the validity of the data scaling law in WAM-Flow.

Different Simulator-Guided GRPO Settings. We analyze the impact of two key design choices in our simulator-guided GRPO framework: group size and reward weighting. As shown in Table 2, varying the group size (number of candidate trajectories sampled per context) reveals a clear trade-off. While smaller groups (size=2) yield marginal gains, a group size of 3 achieves the optimal balance between exploration diversity and training stability, producing the highest PDMS (90.3). Larger groups (size=4) introduce excessive variance, slightly degrading performance.

We further examine the reward function's component weights (EP:TTC:Comfort) in Table 3. Extreme weightings—over-prioritizing either safety (5:20:2) or progress (20:5:2)– suboptimally skew the policy, whereas a balanced ratio (5:5:2) best harmonizes these competing objectives, achieving the superior PDMS of 90.3. This indicates that equitable consideration of ego progress, safety, and comfort is crucial for well-rounded driving performance.

Method	Paradigm	Step	PDMS↑	Infer Time↓	Backbone
FSDrive [50]	AR	-	85.1	10.58s	Qwen2-VL-2B [40]
Epona [52]	AR + Diff.	100	86.2	1.24s	DiT-2.5B [33]
DiffusionDrive [28]	Diff.	2	88.1	0.20s	-
ReCogDrive [27]	AR + Diff.	5	89.6	0.42s	InternVL3-8B [55]
Janus-1.5B [42]	AR	-	-	0.27s	-
Ours	DFM	1	89.1	**0.09s**	Janus-1.5B [42]
		2	89.7	0.19s	
		3	90.0	0.29s	
		5	**90.3**	0.48s	
		10	90.2	0.94s	

Table 6. Intuitive efficiency analysis on NAVSIM.

Coarse-to-fine Sampling Analysis. Table 6 analyzes the coarse-to-fine property of WAM-Flow by varying the number of parallel denoising steps during inference. Increasing the sampling steps from 1 to 5 yields a monotonic improvement in PDMS (89.1 to 90.3), demonstrating that iterative refinement enhances planning quality. Inference time scales approximately linearly with the number of steps, reflecting the parallel nature of the discrete flow matching process. This establishes a flexible trade-off: fewer steps enable faster, coarser plans suitable for real-time constraints, while more steps produce higher-fidelity trajectories.

5. Conclusion

We present WAM-Flow, a vision–language–action model that formulates motion planning as discrete flow matching over a structured token space. The framework incorporates a metric-aligned numerical tokenizer to preserve geometric coherence and employs simulator-guided GRPO to enforce safety and performance in closed-loop control. Evaluated on NAVSIM-v1 and v2 benchmarks, WAM-Flow achieves competitive results, demonstrating its ability to generate high-quality trajectories with a flexible trade-off between inference speed and planning fidelity. This work underscores the potential of discrete flow matching for building reliable and scalable autonomous driving systems.

References

[1] 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. 3

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 2, 7

[3] Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. In *Conference on Robot Learning (CoRL)*, 2025. 6

[4] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024. 7, 8

[5] 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*, pages 26890–26900, 2025. 2, 7

[6] Chaoran Cheng, Jiahan Li, Jiajun Fan, and Ge Liu. α-flow: A unified framework for continuous-state discrete flow matching models. *arXiv preprint arXiv:2504.10283*, 2025. 2

[7] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2022. 7, 8

[8] 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. 3

[9] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 6

[10] Haoge Deng, Ting Pan, Fan Zhang, Yang Liu, Zhuoyan Luo, Yufeng Cui, Wenxuan Wang, Chunhua Shen, Shiguang Shan, Zhaoxiang Zhang, et al. Uniform discrete diffusion with metric path for video generation. *arXiv preprint arXiv:2510.24717*, 2025. 2

[11] 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. *arXiv preprint arXiv:2504.19580*, 2025. 2, 7, 8

[12] 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. 2

[13] 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. 3

[14] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 3

[15] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 3

[16] Zhuolin He, Xinrun Li, Jiacheng Tang, Shoumeng Qiu, Wenfu Wang, Xiangyang Xue, and Jian Pu. Toward camera open-set 3d object detection for autonomous driving scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 26(12):23190–23201, 2025. 1

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[18] Xinmeng Hou, Wuqi Wang, Long Yang, Hao Lin, Jinglun Feng, Haigen Min, and Xiangmo Zhao. Driveagent: Multi-agent structured reasoning with llm and multimodal sensor fusion for autonomous driving. *arXiv preprint arXiv:2505.02123*, 2025. 2

[19] 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. 2, 3

[20] Anqing Jiang, Yu Gao, Zhigang Sun, Yiru Wang, Jijun Wang, Jinghao Chai, Qian Cao, Yuweng Heng, Hao Jiang, Yunda Dong, et al. Diffvla: Vision-language guided diffusion planning for autonomous driving. *arXiv preprint arXiv:2505.19381*, 2025. 2, 3

[21] 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. 3

[22] Bo Jiang, Shaoyu Chen, Qian Zhang, Wenyu Liu, and Xinggang Wang. Alphadrive: Unleashing the power of vlms in autonomous driving via reinforcement learning and reasoning. *arXiv preprint arXiv:2503.07608*, 2025. 3

[23] Amin Karimi Monsefi, Nikhil Bhendawade, Manuel Rafael Ciosici, Dominic Culver, Yizhe Zhang, and Irina Belousova. Fs-dfm: Fast and accurate long text generation with few-step diffusion language models. *arXiv e-prints*, pages arXiv–2509, 2025. 2

[24] Derun Li, Jianwei Ren, Yue Wang, Xin Wen, Pengxiang Li, Leimeng Xu, Kun Zhan, Zhongpu Xia, Peng Jia, Xianpeng Lang, et al. Finetuning generative trajectory model with reinforcement learning from human feedback. *arXiv preprint arXiv:2503.10434*, 2025. 3

[25] Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, Jiayi Liu, Zuxuan Wu, Zhiding Yu, and Jose M Alvarez. Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv preprint arXiv:2503.12820*, 2025. 7, 8

[26] Shufan Li, Konstantinos Kallidromitis, Hritik Bansal, Akash Gokul, Yusuke Kato, Kazuki Kozuka, Jason Kuen, Zhe Lin, Kai-Wei Chang, and Aditya Grover. Lavida: A large diffusion language model for multimodal understanding. *arXiv preprint arXiv:2505.16839*, 2025. 3

[27] 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*, 2025. 1, 2, 5, 7, 8

[28] 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*, pages 12037–12047, 2025. 2, 3, 7, 8

[29] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024. 3

[30] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 2, 5

[31] 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. 3

[32] 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. 3

[33] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 7, 8

[34] Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky TQ Chen. Flow matching with general discrete paths: A kinetic-optimal perspective. *arXiv preprint arXiv:2412.03487*, 2024. 3

[35] Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky TQ Chen. Flow matching with general discrete paths: A kinetic-optimal perspective. In *ICLR*, 2025. 2

[36] Maojiang Su, Mingcheng Lu, Jerry Yao-Chieh Hu, Shang Wu, Zhao Song, Alex Reneau, and Han Liu. A theoretical analysis of discrete flow matching generative models. *arXiv preprint arXiv:2509.22623*, 2025. 2

[37] 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. 3

[38] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 2, 7

[39] Jin Wang, Yao Lai, Aoxue Li, Shifeng Zhang, Jiacheng Sun, Ning Kang, Chengyue Wu, Zhenguo Li, and Ping Luo. Fudoki: Discrete flow-based unified understanding and generation via kinetic-optimal velocities. *arXiv preprint arXiv:2505.20147*, 2025. 2, 3

[40] 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. 2, 7, 8

[41] Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025. 2

[42] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12966–12977, 2025. 2, 7, 8

[43] Shuo Xing, Chengyuan Qian, Yuping Wang, Hongyuan Hua, Kexin Tian, Yang Zhou, and Zhengzhong Tu. Openemma: Open-source multimodal model for end-to-end autonomous driving. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 1001–1009, 2025. 2, 3

[44] 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*, 2025. 1, 2, 3

[45] 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. 3

[46] 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*, 2025. 3

[47] 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. 3

[48] Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *arXiv preprint arXiv:2505.16990*, 2025. 3

[49] Angxiao Yue, Anqi Dong, and Hongteng Xu. Oat-fm: Optimal acceleration transport for improved flow matching. *arXiv preprint arXiv:2509.24936*, 2025. 2

[50] Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, and Xing Wei. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving. *arXiv preprint arXiv:2505.17685*, 2025. 1, 2, 3, 7, 8

[51] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 5

[52] Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, Xun Cao, and Wei Yin. Epona: Autoregressive diffusion world model for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 2, 7, 8

[53] Ruifei Zhang, Junlin Xie, Wei Zhang, Weikai Chen, Xiao Tan, Xiang Wan, and Guanbin Li. Adadrive: Self-adaptive slow-fast system for language-grounded autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5112–5121, 2025. 2

[54] 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*, 2025. 1, 2, 3, 7

[55] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 2, 7, 8