HYBRID-FLOW DRIVE: UNIFYING MASKED DIFFUSION AND CONTINUOUS EMBEDDING FLOW FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

End-to-end autonomous driving with vision-language-action (VLA) models requires trajectories that are accurate, consistent with the observed scene and navigation command, and reliable in long-tail cases. Diffusion language models provide an efficient alternative to purely autoregressive generation, but standard discrete masked-diffusion decoding is not well matched to planning: a fixed `[MASK]` token carries little information about the current noise level, supervision is often limited to corrupted positions, and standard decoding repeatedly discards partial evidence at unresolved tokens. We present Hybrid-Flow Drive, a hybrid flow diffusion VLA that replaces static masks with continuous, noise-aware token embeddings. Our approach has three main innovations: (1) Hybrid Flow Language Model (HFLM), which corrupts response embeddings in continuous space and gives each unresolved position an evolving state; (2) Next-Token-Invisible Bidirectional Attention (NBA), which prevents ground-truth leakage under continuous embeddings and allows dense supervision over response tokens; and (3) Verified Flow Decoding (VFD), which keeps the continuous state across denoising steps and checks the drafted sequence with a causal branch to recover autoregressive consistency. On the Waymo Open Dataset for End-to-End Driving, Hybrid-Flow Drive achieves a highly competitive Rater Feedback Score of 7.772, surpassing prior trajectory-supervised baselines. On nuScenes, it achieves an average L2 trajectory error of 0.31 m. Our results demonstrate that continuous noise-aware states can improve the human-aligned planning quality of diffusion-style end-to-end autonomous driving.

1 INTRODUCTION

Modern end-to-end (E2E) autonomous driving increasingly uses vision-language-action (VLA) models as conditional planners. Given camera observations, ego state and navigation command, a VLA generates structured driving outputs such as scene descriptions and future waypoints (Tian et al., 2025; Zhou et al., 2025; Rowe et al., 2025). This formulation transfers the world knowledge and multimodal reasoning of pretrained vision-language models to planning, which is particularly useful in long-tail events that challenge purely geometric systems (Hu et al., 2023). The generated response must nevertheless remain accurate, consistent with the observed scene and route, and coherent across all fields of the plan.

Many driving VLAs use autoregressive (AR) decoding (Tian et al., 2025; Zhou et al., 2025). Conditioning each token on its complete prefix provides strong token-level fidelity, but it also serializes generation: one new token is committed per step, and an early token cannot be revised after later trajectory evidence becomes available. These properties increase latency and make global consistency across a structured plan dependent on a single left-to-right pass.

Diffusion VLAs mitigate these restrictions by refining multiple response positions in parallel with bidirectional context (Ma et al., 2026; Li et al., 2026). However, standard masked diffusion models (MDMs), including LLaDA-style models (Sahoo et al., 2024; Nie et al., 2025; Ye et al., 2025), assign the same content-free `[MASK]` representation to every unresolved position. The representation therefore does not reveal whether a token is at an early noisy stage or near commitment; training

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.

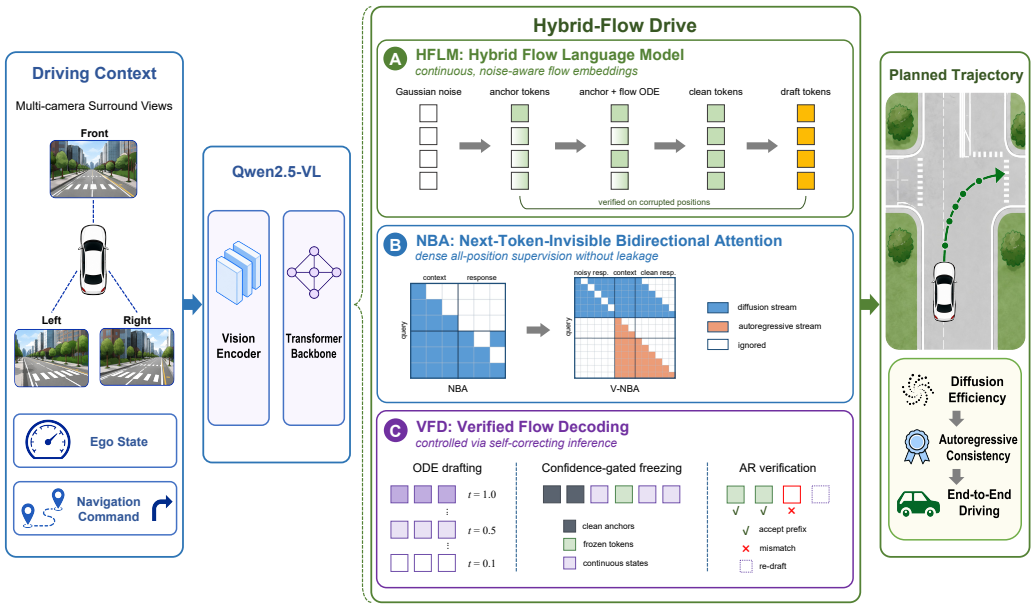


Figure 1: **Overview of Hybrid-Flow Drive.** Given surround-view images, ego state, and navigation command, Hybrid-Flow Drive generates a planned trajectory through three connected modules. HFLM constructs continuous, noise-aware states for editable response tokens; NBA enables dense next-token supervision by hiding each supervised target while retaining bidirectional response context; and VFD transports unresolved states across decoding steps, freezes confident tokens, and verifies the draft with an autoregressive branch.

commonly supervises only masked positions; and iterative decoding does not preserve a graded state for unresolved tokens. Block-diffusion approaches such as Fast-dVLM and Fast-dDrive improve parallelism but retain this static unresolved state (Wu et al., 2026a; Zhang et al., 2026). Fully continuous embedding-space models such as Embedded Language Flows (ELF) instead diffuse the entire language sequence (Hu et al., 2026), while they may depart from the causal next-token interface of pretrained AR backbones, whose world knowledge is what makes VLA planning attractive for long-tail driving.

To this end, we propose Hybrid-Flow Drive (Figure 1), a hybrid flow diffusion VLA addressing the aforementioned limitations. The design is deliberately hybrid rather than fully continuous: only editable response positions carry continuous states, while conditioning and committed tokens remain clean anchors and targets are still predicted discretely under cross-entropy. Therefore, Hybrid-Flow Drive is continuous where masked diffusion methods are static, and discrete where ELF is continuous, which preserves the token-level fidelity that a structured driving plan requires. The key contributions are:

- We propose the **Hybrid Flow Language Model (HFLM)**, which replaces the discrete `[MASK]` token with a continuous-time interpolant between Gaussian noise and the ground-truth embedding. Coupling the flow time to the corruption ratio gives the input an explicit notion of noise level and yields a smoother training curriculum.
- We introduce **Next-Token-Invisible Bidirectional Attention (NBA)**, a visibility pattern that hides only the immediate next token from each predictor. This prevents target leakage while allowing all response positions to be supervised in one pass, and it preserves both diffusion-style bidirectional reasoning over the response and the causal prompt topology of the pretrained AR backbone.
- We develop **Verified Flow Decoding (VFD)**, which keeps a continuous flow state for uncommitted tokens, updates that state with Euler ordinary differential equation (ODE) drafting steps, and then verifies the draft with a causal branch. The decoder can therefore reuse earlier refinement rather than making isolated commitments at each step.

The complete system decodes $7.33\times$ faster than the autoregressive baseline, achieves a Rater Feedback Score (RFS) of 7.772 on WOD-E2E and an average L2 error of 0.31 m on nuScenes, providing both controlled evidence and cross-dataset validation for the proposed continuous unresolved states.

2 Related Work

Vision-Language-Action Models for Driving. Autoregressive VLAs couple language-model reasoning with trajectory prediction (Tian et al., 2025), inheriting strong sequence modeling and multimodal reasoning from pretrained vision-language models (VLMs). Their causal factorization, however, commits one token at a time and cannot revise an earlier field using a later trajectory field. Diffusion driving VLAs (Ma et al., 2026; Li et al., 2026) instead refine multiple response positions with bidirectional context, improving parallelism and cross-field interaction. Few works, however, examine the masked-diffusion core itself, where static mask representations, sparse supervision over corrupted positions, and memoryless decoding can all affect planning quality.

Diffusion Large Language Models. Discrete diffusion for text spans foundational formulations (Austin et al., 2021), masked objectives (Sahoo et al., 2024), and large-scale models such as LLaDA and Dream (Nie et al., 2025; Ye et al., 2025). Block Diffusion and Fast-dLLM combine intra-block bidirectional attention with cross-block causality (Arriola et al., 2025; Wu et al., 2026b); Fast-dVLM and Fast-dDrive adapt block diffusion and draft verification to multimodal generation and driving (Wu et al., 2026a; Zhang et al., 2026). These methods improve parallel decoding, but unresolved positions are still represented by a shared discrete `[MASK]` token, and supervision is typically restricted to masked targets. Our HFLM gives each unresolved position a time-dependent continuous state, while NBA changes the visibility mask so that every response target can be supervised without leakage.

Continuous Corruption for Language. Flow matching (Lipman et al., 2023) models continuous transport between noise and data distributions and has been effective for generation in continuous modalities. For text, Discrete Flow Matching (Gat et al., 2024) defines continuous-time Markov chains over discrete states, Diffusion-LM and ELF perturb continuous word embeddings (Li et al., 2022; Hu et al., 2026), and Flow Map Language Models learn continuous denoising maps (Lee et al., 2026). Continuous corrupted embeddings provide two properties that a shared mask cannot: the input can encode corruption severity, and an unresolved position can retain a graded estimate across refinement steps. Our use of this representation is deliberately hybrid. Compared with fully continuous embedding-space language models, HFLM corrupts only editable response positions and retains cross-entropy token reconstruction, confidence-based hard commitment, and an AR branch for verification. Unlike CADD (Zheng et al., 2026), which pairs a discrete process with an auxiliary continuous latent process, Hybrid-Flow Drive maintains one continuous corrupted state directly in token-embedding space, and the transport toward the current clean-token estimate is then analytic rather than a learned latent dynamics, adding no parameters or extra forward passes beyond the backbone itself.

Iterative Generation and Draft Verification. Unified multimodal generators such as Show-o generate discrete image tokens through MaskGIT-style iterative masking and denoising (Xie et al., 2025). That process is designed for a discrete image-token grid: unresolved image positions are represented by mask tokens and are repeatedly selected for discrete reconstruction. VFD instead decodes a structured language-action response, preserves a continuous embedding for every unresolved position across steps, and uses a causal language branch to verify the longest consistent token prefix. Speculative and self-speculative decoding introduced this draft-and-verify pattern for AR acceleration (Leviathan et al., 2023; Zhang et al., 2024), and diffusion language models later adapted it to parallel drafts (Wu et al., 2026b;a). VFD differs from these methods in the draft state: rejected or undecided positions retain their accumulated continuous evidence rather than returning to a shared static mask.

3 METHOD

During training, HFLM samples a corruption time t, replaces each selected editable response embedding with a continuous noisy state, and assembles the resulting embedding matrix $\mathbf{H}_t$ together with clean conditioning anchors. NBA then hides the target token from its one-step-shifted predictor and applies cross-entropy supervision to every response target in a single forward pass. During inference, VFD initializes editable response positions from Gaussian noise, transports their states toward the current clean-token estimates, freezes confident tokens, and invokes the same model's causal AR branch to verify the draft. The following subsections define the baseline masked-diffusion objective, the continuous representation and training objective, the leakage-free attention mask, and the verified decoder.

3.1 MASKED DIFFUSION MODEL (MDM)

Let $\mathbf{x}_1 = (x_1^1, \ldots, x_1^L)$ be a target token sequence sampled from $p_{\text{data}}(\mathbf{x}_1)$. Let $\mathbf{c}$ denote the conditioning sequence formed from visual features, ego/navigation inputs, and the text prompt. We use the subscript 1 for clean data to align the masked-diffusion and continuous-corruption directions below. Accordingly, forward corruption runs from $t = 1$ (clean data) to $t = 0$ (fully corrupted), whereas reverse denoising runs from $t = 0$ to $t = 1$.

A masked diffusion model (Sahoo et al., 2024) defines a forward process that progressively replaces tokens with a special `[MASK]` token according to a noise schedule, yielding a corrupted sequence $\mathbf{x}_t$. The reverse process predicts replacements for masked positions while keeping visible tokens fixed, minimizing the masked cross-entropy:

$$\mathcal{L}_{\text{MDM}}(\theta) := \mathbb{E}_{t,\mathbf{x}_1,\mathbf{x}_t}\left[-\frac{1}{|\mathcal{M}_t|}\sum_{i\in\mathcal{M}_t}\log p_\theta\big(x_1^i \mid \mathbf{x}_t, \mathbf{c}\big)\right], \qquad \mathcal{M}_t = \{i : x_t^i = \texttt{[MASK]}\}. \tag{1}$$

Here, i is a response position, $\mathcal{M}_t$ is the set of corrupted positions, and p_θ is the token distribution obtained by applying the language-model head to the transformer's output at the corresponding prediction position. Thus, Eq. (1) supervises target token x_1^i only when position i is corrupted. Section 3.3 specifies the shifted prediction positions used by NBA to extend this supervision to every response target.

3.2 HYBRID FLOW LANGUAGE MODEL (HFLM)

Unlike fully continuous embedding-space language models such as ELF (Hu et al., 2026), HFLM corrupts only editable response positions and keeps conditioning and committed tokens as clean anchors. From the MDM perspective, HFLM replaces the shared `[MASK]` embedding with a continuous corrupted embedding whose value changes with noise level. The model still predicts discrete target tokens with cross-entropy. Figure 2 summarizes this hybrid construction.

Forward Corruption Process. Let $E(\cdot)$ be the token embedding layer and let $\mathbf{e}_1^i = E(x_1^i) \in \mathbb{R}^d$ be the embedding of target token x_1^i. We associate every editable position with a scalar time $t \in [0, 1]$, where $t = 0$ is the Gaussian-noise endpoint and $t = 1$ is the low-noise data-side endpoint. Given $\mathbf{x}_1$, each editable position is included in the corruption set $\mathcal{M}_t \subseteq \mathcal{E}$ with probability $1 - t$. Thus, smaller t yields more corrupted positions and weaker token evidence.

For each $i \in \mathcal{M}_t$, sample an independent noise vector $\boldsymbol{\epsilon}^i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and form:

$$\begin{aligned}\hat{\mathbf{e}}_0^i &= \boldsymbol{\epsilon}^i,\\ \hat{\mathbf{e}}_1^i &= \gamma\boldsymbol{\epsilon}^i + (1-\gamma)\mathbf{e}_1^i,\\ \hat{\mathbf{e}}_t^i &= t\hat{\mathbf{e}}_1^i + (1-t)\hat{\mathbf{e}}_0^i.\end{aligned} \tag{2}$$

The scalar $0 < \gamma < 1$ retains a small amount of noise at the data-side endpoint, preventing near-identity reconstruction at large t. The response embedding matrix is:

$$\mathbf{H}_t = \begin{bmatrix}(\mathbf{h}_t^1)^\top\\ \vdots\\ (\mathbf{h}_t^L)^\top\end{bmatrix}, \qquad \mathbf{h}_t^i = \begin{cases}\hat{\mathbf{e}}_t^i, & i \in \mathcal{M}_t,\\ \mathbf{e}_1^i, & i \notin \mathcal{M}_t.\end{cases} \tag{3}$$

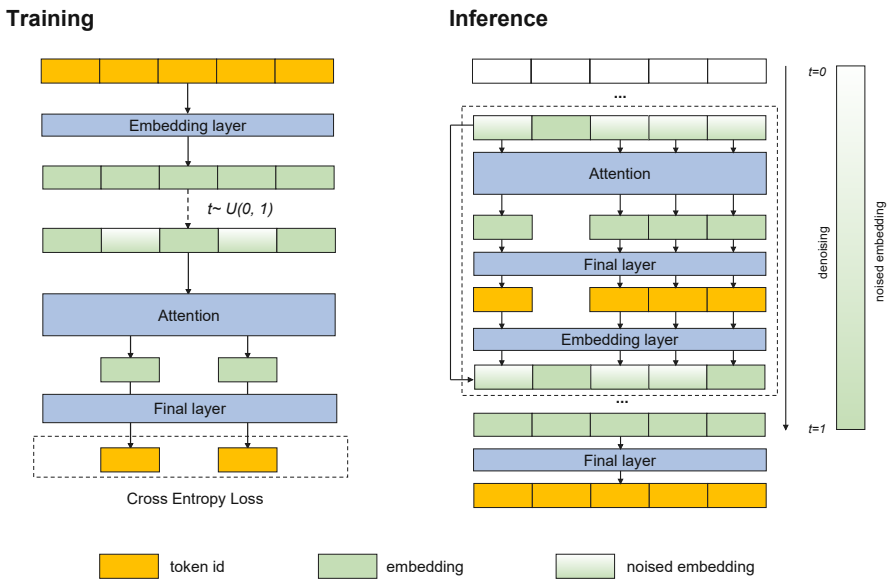


Figure 2: **HFLM training and inference.** Training replaces editable response embeddings with continuous corrupted states while keeping anchors clean. Inference transports unresolved states from noise toward current clean-token estimates, so partial evidence persists instead of returning to a static mask.

Therefore, row i of $\mathbf{H}_t$ is a continuous corrupted vector when the response position is unresolved and a clean anchor otherwise. Equations (2)–(3) define the fixed corruption distribution $q(\mathbf{H}_t \mid \mathbf{x}_1)$.

Hybrid Flow Reconstruction Objective. HFLM is trained to reconstruct target tokens from the corrupted embedding matrix:

$$\mathcal{L}_{\mathrm{HFLM}} = -\mathbb{E}_{\mathbf{x}_1}\mathbb{E}_{\mathbf{H}_t|\mathbf{x}_1}\left[\ell_{\mathrm{HFLM}}(\mathbf{x}_1, \mathbf{H}_t)\right],$$
$$\ell_{\mathrm{HFLM}}(\mathbf{x}_1, \mathbf{H}_t) = \frac{1}{|\mathcal{M}_t|}\sum_{i\in\mathcal{M}_t} \log p_\theta\left(x_1^i \mid \mathbf{H}_t, \mathbf{c}\right). \tag{4}$$

Here, p_θ is the vocabulary distribution produced by the transformer and its language-model head, and the summand for position i reconstructs target token x_1^i. This cross-entropy is a reconstruction objective under a fixed continuous corruption process; it can be related to the reconstruction term of a fixed-corruption variational objective, but we do not train a separate velocity field. NBA in Section 3.3 specifies the exact shifted logits and extends Eq. (4) from corrupted positions to all response targets.

Sampling t controls the curriculum over corruption levels. Low-t examples train recovery from weak semantic evidence, whereas high-t examples train refinement near a target embedding. All HFLM ablations use the same time distribution so that they isolate representation, attention, and decoding changes.

Straight-Line Denoising Transport. At inference, every editable response state is initialized independently from $\mathcal{N}(\mathbf{0}, \mathbf{I})$. We use the increasing grid $0 = t_0 < t_1 < \cdots < t_N = 1$, with $\Delta t_k = t_{k+1} - t_k$, while anchors retain clean embeddings. At step k, the model and language-model head produce vocabulary logits at the NBA prediction positions. Their argmax tokens form the current clean-token estimate $\bar{\mathbf{x}}_1$, with $\bar{\mathbf{e}}_1^i = E(\bar{x}_1^i)$. For each undecoded position $i \in \mathcal{U}_k$, we use the straight-line velocity toward this estimate:

$$\mathbf{v}_k^i = \frac{\bar{\mathbf{e}}_1^i - \hat{\mathbf{e}}_{t_k}^i}{1 - t_k},$$
$$\hat{\mathbf{e}}_{t_{k+1}}^i = \hat{\mathbf{e}}_{t_k}^i + \mathbf{v}_k^i \Delta t_k, \qquad i \in \mathcal{U}_k. \tag{5}$$

The denominator is nonzero because the update is evaluated only for $k < N$. Equation (5) is an analytic Euler transport induced by the interpolant, rather than a learned velocity-matching objective. After each update, VFD commits positions whose confidence exceeds τ, replaces their rows in $\mathbf{H}_{t_{k+1}}$ with clean token embeddings, and retains the updated continuous rows for all remaining positions.

3.3 Next-Token-Invisible Bidirectional Attention

Dense Supervision under Non-Autoregressive Training. We initialize HFLM from a pretrained autoregressive model backbone and continue training, preserving the linguistic and multimodal capabilities of the checkpoint. The adapted objective should remain compatible with next-token prediction (NTP). Under NTP, the representation at a response position predicts the following response token. Naively making the response block fully bidirectional creates target leakage: a predictor can directly attend to the clean embedding of the next token that it is trained to predict. Conventional MDMs avoid this leakage by supervising only masked targets, which leads to a sparse objective. HFLM requires a stricter construction because even a corrupted embedding contains graded information about its clean token.

Next-Token-Invisible Visibility Pattern. Let the conditioning context contain M tokens and the response contain L tokens. In the concatenated sequence

$$\mathbf{z} = (c^1, \ldots, c^M, x_1^1, \ldots, x_1^L),$$

global positions $1{:}M$ are context positions and $M{+}1{:}M{+}L$ are response positions. For a query at global position i and a key/value at global position j, NBA uses the visibility matrix $\mathbf{A} \in \{0,1\}^{(M+L)\times(M+L)}$, whose scalar entries are:

$$A_{ij} = \begin{cases} 1, & i \le M \text{ and } j \le i, \\ 1, & i > M \text{ and } j \le M, \\ 1, & i > M,\ M < j \le M+L, \text{ and } j \ne i+1, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The context block therefore remains causal, matching Qwen-style autoregressive VLMs. A response query attends to the full context and all response positions except its next token.

More precisely, target token x_1^k uses predictor position

$$r_k = M + k - 1, \qquad k \in \{1, \ldots, L\}.$$

For $k = 1$, $r_1 = M$ is the last context position; for $k > 1$, r_k is the position of the preceding local response token x_1^{k-1}. Thus, a response query at global position i predicts the token at global position $i+1$, and Eq. (6) hides exactly that target key/value. A detailed visualization of NBA and its verification-augmented variant is provided in Appendix A.

Dense NTP Objective. Let $\mathbf{C} \in \mathbb{R}^{M\times d}$ be the context embedding matrix and $\mathbf{Z}_t = [\mathbf{C}; \mathbf{H}_t]$ be the full input embedding matrix. The transformer produces $\mathbf{O}_\theta = F_\theta(\mathbf{Z}_t; \mathbf{A})$. If $\mathbf{o}_{\theta,r_k}$ is row r_k of $\mathbf{O}_\theta$ and $\mathbf{W}_{\mathrm{lm}}$ is the language-model head, then the logit vector for target x_1^k and its probability are:

$$\begin{aligned} \boldsymbol{\ell}_{\theta,k} &= \mathbf{W}_{\mathrm{lm}} \mathbf{o}_{\theta,r_k}, \\ p_\theta(x_1^k \mid \mathbf{H}_t, \mathbf{c}; \mathbf{A}) &= \mathrm{softmax}(\boldsymbol{\ell}_{\theta,k})_{x_1^k}. \end{aligned} \tag{7}$$

Consequently, NBA trains every response target with the one-token-shifted objective:

$$\begin{aligned} \mathcal{L}_{\mathrm{NBA}} &= -\mathbb{E}_{\mathbf{x}_1} \mathbb{E}_{\mathbf{H}_t|\mathbf{x}_1} \left[\ell_{\mathrm{NBA}}(\mathbf{x}_1, \mathbf{H}_t) \right], \\ \ell_{\mathrm{NBA}}(\mathbf{x}_1, \mathbf{H}_t) &= \frac{1}{L} \sum_{k=1}^{L} \log p_\theta\big(x_1^k \mid \mathbf{H}_t, \mathbf{c}; \mathbf{A}\big). \end{aligned} \tag{8}$$

Here, the k-th summand uses the logits at predictor r_k to supervise target x_1^k. Unlike Eqs. (1) and (4), the sum is not restricted to $\mathcal{M}_t$; corrupted and clean response positions both contribute. The target is hidden by construction, while the remaining response positions provide bidirectional context. NBA therefore supplies a practical interface between dense AR-style training and parallel HFLM decoding.

Algorithm 1 Verified Flow Decoding (VFD).

Require: context $\mathbf{c}$; anchor set $\mathcal{A}$; denoising steps N; confidence threshold τ.
Ensure: response tokens $\hat{\mathbf{x}}_1$.

```
Initialize editable response rows from Gaussian noise and place anchors as clean embeddings.
while undecoded set 𝒰 is nonempty do
    for k = 0 to N − 1 do
        Run the model to obtain clean-token estimates and confidences.
        Update undecoded rows with Eq. (5).
        Freeze positions whose confidence exceeds τ.
    end for
    Verify the current draft with the causal AR branch.
    Accept the matching prefix and commit the verifier token at the first mismatch.
    Retain subsequent positions as undecoded continuous states.
end while
Return the committed response x̂_1.
```

3.4 Verified Flow Decoding

At inference, Hybrid-Flow Drive drafts tokens through straight-line transport via Euler ordinary differential equation (ODE), stabilizes the draft by freezing confident positions, and verifies it with a causal branch.

ODE Drafting and Confidence-Gated Freezing. Let $\mathcal{E}$ be the editable response positions and let $\mathcal{U}_k \subseteq \mathcal{E}$ be the positions that remain undecoded at step k. Equation (5) maintains a coherent denoising trajectory by updating each row $\hat{\mathbf{e}}^i_{t_k}$ toward the current clean-token estimate. After the update, the model assigns confidence $s^i_k = \max_v \mathrm{softmax}(\boldsymbol{\ell}_{\theta,i})_v$ to position i. If $s^i_k > \tau$, VFD commits the predicted token, replaces the row by its clean embedding, and removes i from $\mathcal{U}_k$. The embedding matrix therefore contains three types of rows: clean conditioning anchors, frozen committed tokens, and continuous states for unresolved response positions. Unlike MaskGIT-style discrete generation, an unresolved row is not reset to a mask token after each round.

Autoregressive Verification. Parallel drafting relaxes the left-to-right factorization and can reduce token consistency. AR-verify (Wu et al., 2026a) re-examines a draft with a causal branch: it accepts the longest prefix that agrees with the AR prediction and uses the verifier's token to correct the first disagreement. We support this operation by extending NBA to Verification-Augmented Next-Token-Invisible Bidirectional Attention (V-NBA). Appendix A specifies its two streams and visibility matrix. Both streams share model parameters: the HFLM stream drafts in parallel, while the AR stream conditions on the accepted clean prefix.

During VFD, anchors pass automatically, matching draft tokens are accepted, and the first mismatch is replaced by one verifier token. Positions after that token remain undecoded and continue from their accumulated continuous states in the next drafting round. Every verification round commits either the full matching remainder or at least the verifier token at the first mismatch. Thus, the undecoded set strictly shrinks and the outer loop terminates after at most L verification rounds. Algorithm 1 summarizes VFD.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. We evaluate Hybrid-Flow Drive on two driving benchmarks. The Waymo Open Dataset for End-to-End Driving (WOD-E2E) (Xu et al., 2026) targets long-tail scenarios and provides 360° camera views, ego state, and high-level routing. Its primary metric is the Rater Feedback Score (RFS), a human-aligned open-loop score in $[0, 10]$ that compares a predicted 5-s trajectory with three rater-annotated references inside speed-scaled trust regions. We also report average displacement error (ADE) at 3-s and 5-s horizons. In addition, we report an additional cross-dataset evaluation on nuScenes (Caesar et al., 2020) in Appendix C.

Table 1: Comparison on the WOD-E2E test set. Metrics for reference methods are taken from publicly available materials. CoT denotes chain-of-thought reasoning, and Traj denotes trajectory output. Best in **bold**.

Method	Paradigm	Output	RFS ↑	ADE@5s ↓	ADE@3s ↓
MTR-VP (Keskar et al., 2025)	Vision Transformer	Traj	7.343	3.349	1.423
EMMA (Hwang et al., 2025)	AR VLM	Traj	7.528	3.018	1.320
AutoVLA (Zhou et al., 2025)	AR VLM	CoT+Traj	7.557	2.958	1.351
dVLM-AD (Ma et al., 2026)	Diffusion VLM	CoT+Traj	7.633	3.022	1.285
DiffusionLTF (Nguyen et al., 2025)	Diffusion Policy	Traj	7.717	**2.891**	1.356
Hybrid-Flow Drive (Ours)	HFLM	Traj	**7.772**	2.972	**1.254**

Table 2: Controlled comparison under a unified training recipe, data format, and model scale. TPSec denotes tokens per second, and TPStep denotes the average effective tokens committed per step. Best in **bold**.

Method	RFS ↑	ADE@5s ↓	ADE@3s ↓	TPSec ↑	TPStep ↑
AR Baseline (vanilla Qwen2.5-VL-3B)	7.500	3.089	1.307	45.73	1.00
MDM Baseline (dVLM-AD style)	7.661	3.054	1.303	106.81	3.56
Block Diffusion (Fast-dVLM style)	7.555	3.092	1.320	99.67	3.60
Hybrid-Flow Drive ($N = 1$)	**7.772**	**2.975**	**1.254**	**335.30**	**10.33**

Backbone and Implementation. Hybrid-Flow Drive fine-tunes Qwen2.5-VL-3B-Instruct (Bai et al., 2025) to predict driving trajectories without chain-of-thought supervision. Complete optimization, compute, input, and decoding configurations are provided in Appendix B.

4.2 Main Results

Table 1 compares Hybrid-Flow Drive with representative AR, diffusion, and trajectory-policy systems. Because the public entries differ in output format and training recipe, this is an external benchmark comparison rather than a controlled component study. Under this external comparison setting, Hybrid-Flow Drive achieves an RFS of 7.772, a favorable ADE@3s of 1.254, and a competitive ADE@5s of 2.972. These results suggest that continuous unresolved states are useful for human-aligned trajectory planning, while the controlled study below isolates the model-level gain.

To complement the comparison with external benchmarks, we conduct a more direct evaluation of the gains brought by Hybrid-Flow Drive against autoregressive, masked-diffusion, and block-diffusion baselines. All baselines are trained under the same recipe, data format, and model scale. The autoregressive baseline uses vanilla Qwen2.5-VL-3B (Bai et al., 2025). The masked-diffusion baseline follows the dVLM-AD design (Ma et al., 2026), while the block-diffusion baseline follows Fast-dVLM (Wu et al., 2026a) with a block size of 32. All systems use the same WOD-E2E setup and the trajectory is the sole supervision signal. In addition to RFS and ADE, we measure inference efficiency using TPSec, defined as the number of generated tokens per second, and TPStep, defined as the average number of effective tokens committed at each decoding step. Under this definition, TPStep is exactly 1 for standard autoregressive decoding. Efficiency benchmarks are evaluated on a single NVIDIA H200 GPU.

Table 2 shows that the advantage of Hybrid-Flow Drive remains clear after controlling for major confounding factors. Compared with the vanilla autoregressive baseline, Hybrid-Flow Drive improves RFS from 7.500 to 7.772 and reduces ADE@5s from 3.089 to 2.975, while increasing throughput from 45.73 to 335.30 tokens per second. The improvement is not limited to autoregressive decoding, as Hybrid-Flow Drive also achieves better accuracy and speed than the masked-diffusion and block-diffusion baselines. Even with only one ODE drafting step, its performance is quite similar to that of the four-step setting reported in Table 1, showing the promising capability of the straight-line denoising strategy. At the same time, its TPStep of 10.33 shows that ODE drafting combined with autoregressive verification can commit substantially more valid trajectory tokens at each decoding step.

Table 3: Incremental module ablation on WOD-E2E val. Best in **bold**.

HFLM	NBA	VFD	ADE@3s↓	ADE@5s↓
			0.6920	1.8454
✓			0.6692	1.8096
✓	✓		0.6137	1.6804
✓		✓	0.4867	1.5070
✓	✓	✓	**0.4604**	**1.4223**

4.3 Ablation Studies

We isolate the two components that Hybrid-Flow Drive adds on top of the HFLM backbone. All ablations train on a 90k-frame subset randomly sampled from the full training set and report ADE on a held-out mini validation set. We toggle NBA and VFD modules. With NBA disabled, the model trains with standard causal attention; with VFD disabled, decoding reverts to plain iterative denoising without confidence-gated freezing or autoregressive verification. The first row in Table 3 is the standard causal AR baseline, and the second row enables HFLM alone.

Empirically, the reported numbers show that both components help and that they are complementary to each other. VFD is the larger single factor: adding it to the HFLM backbone lowers ADE@3s from 0.6692 to 0.4867 and ADE@5s from 1.8096 to 1.5070, because ODE drafting plus autoregressive verification corrects the greedy commitments that a plain iterative decoder makes. NBA helps on its own ($0.6692 \rightarrow 0.6137$) and adds a further gain on top of VFD ($0.4867 \rightarrow 0.4604$), as the leakage-free bidirectional context and dense supervision it provides yield a stronger draft and a better-aligned verifier. Using both gives the best result, reducing ADE@3s by 31% and ADE@5s by 21% relative to the HFLM-only configuration.

5 Conclusion

We introduced Hybrid-Flow Drive, a hybrid flow diffusion VLA for end-to-end autonomous driving. HFLM replaces static `[MASK]` tokens with continuous, noise-aware response states, NBA provides dense leakage-free next-token supervision, and VFD preserves unresolved states while verifying drafts with a causal branch. Together, the modules address static representation, sparse supervision, and memoryless decoding in masked-diffusion VLAs.

On WOD-E2E, Hybrid-Flow Drive achieves an RFS of 7.772 and a favorable ADE@3s of 1.254 among the compared methods. Under controlled settings, it decodes $7.33\times$ faster than the AR baseline and $3.14\times$ faster than the discrete MDM baseline while improving RFS. On nuScenes, it achieves an average L2 error of 0.31 m. The ablation further shows complementary gains from NBA and VFD, which together reduce ADE@3s and ADE@5s relative to HFLM alone.

Hybrid-Flow Drive is not without its limitations. The evaluation remains open-loop, and the nuScenes experiment uses only three consecutive front-camera frames. Closed-loop testing, broader sensor configurations, and longer-horizon evaluation are therefore important next steps. These limitations do not affect the controlled comparison of decoding paradigms, but they bound claims about deployment-level driving performance.

Reproducibility Statement

The model architecture and training objective are specified in Sections 3.2–3.4. Appendix B reports the optimization, data, compute, and decoding configurations used in our experiments, and Appendix C provides the complete nuScenes evaluation protocol and results.

AI use statement

We used generative AI tools to assist with non-substantive software code editing, improve the readability of the manuscript, and check compliance with submission formatting requirements. We did not use generative AI tools to generate synthetic data, develop theoretical or conceptual frameworks,

formulate mathematical claims or proofs, propose research hypotheses, design the research methodology or experiments, implement the proposed method, process datasets, conduct data analysis, or interpret experimental results. All AI-assisted code changes were manually reviewed and tested by the authors. All AI-edited text was checked for technical accuracy and consistency with the intended claims, and all formatting suggestions were verified against the official ICLR 2027 guidelines and the compiled manuscript. The authors take responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI.

REFERENCES

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. In *International Conference on Learning Representations (ICLR)*, 2025.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11621–11631, 2020.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Keya Hu, Linlu Qiu, Yiyang Lu, Hanhong Zhao, Tianhong Li, Yoon Kim, Jacob Andreas, and Kaiming He. ELF: Embedded language flows. *arXiv preprint arXiv:2605.10938*, 2026.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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, Yin Zhou, James Guo, Dragomir Anguelov, and Mingxing Tan. EMMA: End-to-end multimodal model for autonomous driving. *Transactions on Machine Learning Research*, 2025.

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 (ICCV)*, pp. 8340–8350, 2023.

Maitrayee Keskar, Mohan Trivedi, and Ross Greer. MTR-VP: Towards end-to-end trajectory planning through context-driven image encoding and multiple trajectory prediction. *arXiv preprint arXiv:2511.22181*, 2025.

Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M. Boffi, and Jinwoo Kim. Flow Map Language Models: One-step language modeling via continuous denoising. *arXiv preprint arXiv:2602.16813*, 2026.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning (ICML)*, pp. 19274–19286, 2023.

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. In *International Conference on Learning Representations (ICLR)*, 2026.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S. Liang, and Tatsunori B. Hashimoto. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Yingzi Ma, Yulong Cao, Wenhao Ding, Shuibai Zhang, Yan Wang, Boris Ivanovic, Ming Jiang, Marco Pavone, and Chaowei Xiao. dVLM-AD: Enhance diffusion vision-language-model for driving via controllable reasoning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, 2026.

Long Nguyen, Micha Fauth, Bernhard Jaeger, Daniel Dauner, Maximilian Igl, Andreas Geiger, and Kashyap Chitta. Open X-AV: Unifying end-to-end autonomous driving datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2025.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Luke Rowe, Rodrigue de Schaetzen, Roger Girgis, Christopher Pal, and Liam Paull. Poutine: Vision-language-trajectory pre-training and reinforcement learning post-training enable robust end-to-end autonomous driving. *arXiv preprint arXiv:2506.11234*, 2025.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Mariano Marroquin, Justin T. Chiu, Alexander M. Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 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. In *Proceedings of the 8th Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4698–4726. PMLR, 2025.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22442–22452, 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, 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. In *International Conference on Learning Representations (ICLR)*, 2026b.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *International Conference on Learning Representations (ICLR)*, 2025.

Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Kate Tolstaya, Sarah Tang, Brandyn White, Ben Sapp, Mingxing Tan, Jyh-Jing Hwang, and Dragomir Anguelov. WOD-E2E: Waymo open dataset for end-to-end driving in challenging long-tail scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3709–3718, 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*, 2025.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. Draft & Verify: Lossless large language model acceleration via self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 11263–11282, 2024.

Kewei Zhang, Jin Wang, Sensen Gao, Chengyue Wu, Yulong Cao, Songyang Han, Boris Ivanovic, Langechuan Liu, Marco Pavone, Song Han, Daquan Zhou, and Enze Xie. Fast-dDrive: Efficient block-diffusion VLM for autonomous driving. *arXiv preprint arXiv:2605.23163*, 2026.

Huangjie Zheng, Shansan Gong, Ruixiang Zhang, Tianrong Chen, Jiatao Gu, Mingyuan Zhou, Navdeep Jaitly, and Yizhe Zhang. Continuously augmented discrete diffusion model for categorical generative modeling. In *International Conference on Learning Representations (ICLR)*, 2026.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, Volker Tresp, and Alois Knoll. OpenDriveVLA: Towards end-to-end autonomous driving with large vision language action model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(16):13782–13790, 2026.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

A HFLM WITH AR VERIFICATION

To equip HFLM with AR verification, each training instance contains aligned noisy-response and clean-response streams around the conditioning context. Figure 3 visualizes the corresponding NBA and V-NBA visibility masks before we formalize the joint input and loss below.

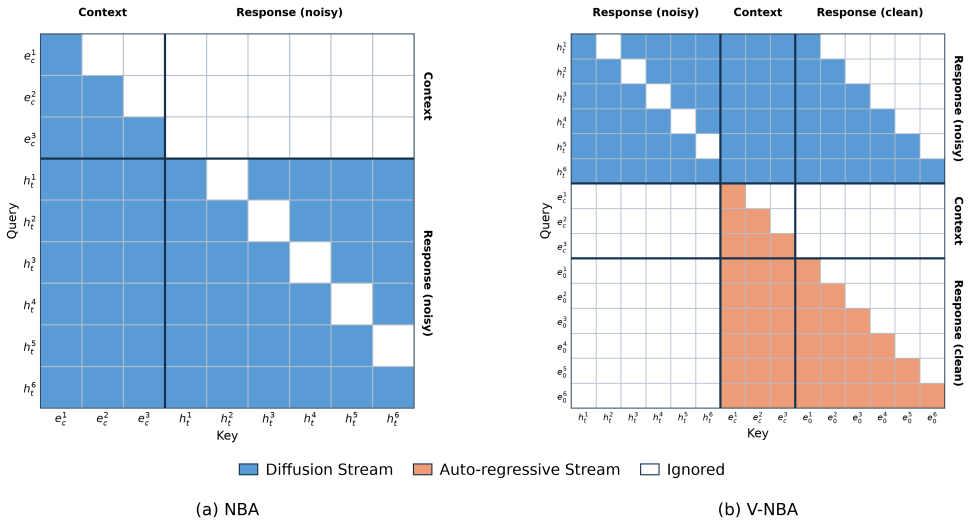


Figure 3: **Detailed visibility masks for NBA and V-NBA.** Rows are query positions and columns are key/value positions. NBA keeps the context causal and hides the next response target from each noisy-response predictor. V-NBA adds a clean-response causal stream for AR verification while preserving the next-token-invisible noisy stream for parallel drafting.

Each training instance contains three consecutive embedding blocks:

$$\mathbf{Z}_t = \begin{bmatrix} \mathbf{H}_t \\ \mathbf{C} \\ \mathbf{E}_1 \end{bmatrix}. \tag{9}$$

Here, $\mathbf{H}_t \in \mathbb{R}^{L\times d}$ is the noisy-response embedding matrix, $\mathbf{C} \in \mathbb{R}^{M\times d}$ is the context matrix, and $\mathbf{E}_1 \in \mathbb{R}^{L\times d}$ is the clean-response embedding matrix. Although the noisy response is physically placed before the context, its position identifiers are shared with the clean response:

$$\mathrm{Pos}(c^m) = m, \qquad \mathrm{Pos}(\mathbf{h}_t^k) = \mathrm{Pos}(\mathbf{e}_1^k) = M + k. \tag{10}$$

Thus, the two response streams are aligned in positional space, while the context retains the position range $1{:}M$.

We extend NBA to V-NBA, as shown in Figure 3. The AR stream (orange) uses causal attention over context and preceding clean response tokens, so it can verify a draft without an additional model. The HFLM stream (blue) attends to context, the noisy response under the next-token-invisible constraint, and only the clean-response prefix allowed by the mask. Consequently, the predictor aligned with target x_1^k cannot observe the target's clean embedding.

Following Fast-dVLM (Wu et al., 2026a), both streams share one forward pass. Let $\mathbf{A}_{\text{V-NBA}}$ be the visibility matrix and define:

$$\mathbf{O}_\theta = F_\theta(\mathbf{Z}_t; \mathbf{A}_{\text{V-NBA}}). \tag{11}$$

For target x_1^k, the HFLM distribution p_θ^{N} is the softmax of the language-model logits at its NBA-aligned noisy-stream predictor, and p_θ^{C} is the softmax at the standard one-token-shifted clean-stream predictor. Context logits are not supervised. For compactness, write $\mathbf{A}_{\mathrm{V}} = \mathbf{A}_{\text{V-NBA}}$. The joint loss is:

$$\mathcal{L}_{\text{V-NBA}} = \mathcal{L}_{\text{HFLM}} + \beta\mathcal{L}_{\text{AR}}, \tag{12}$$

where

$$\mathcal{L}_{\mathrm{HFLM}} = -\mathbb{E}_{\mathbf{x}_1}\mathbb{E}_{\mathbf{H}_t|\mathbf{x}_1}\left[\frac{1}{L}\sum_{k=1}^{L}\log p_\theta^{\mathrm{N}}\left(x_1^k \mid \mathbf{Z}_t; \mathbf{A}_{\mathrm{V}}\right)\right], \tag{13}$$

and

$$\mathcal{L}_{\mathrm{AR}} = -\mathbb{E}_{\mathbf{x}_1}\left[\frac{1}{L}\sum_{k=1}^{L}\log p_\theta^{\mathrm{C}}\left(x_1^k \mid \mathbf{C}, \mathbf{x}_1^{<k}; \mathbf{A}_{\mathrm{V}}\right)\right]. \tag{14}$$

The scalar β balances the parallel HFLM branch and causal AR branch. At inference, the first branch drafts multiple tokens, and the second accepts the longest consistent prefix and supplies the first correction.

B Implementation Details

Backbone and Optimization. We initialize Hybrid-Flow Drive from the Qwen2.5-VL-3B-Instruct checkpoint (Bai et al., 2025) and update all model parameters rather than using parameter-efficient adapters. WOD-E2E and nuScenes use the same optimization recipe in Table 4. We train both configurations for three epochs with AdamW in BF16 precision. The warmup occupies the first 3% of training, after which the learning rate remains constant. Gradients are clipped with a threshold of 1.0.

Table 4: Shared training hyperparameters for WOD-E2E and nuScenes.

Hyperparameter	Value
Backbone	Qwen2.5-VL-3B-Instruct
Fine-tuning strategy	Full-parameter fine-tuning
Optimizer	AdamW
Optimizer betas (β_1, β_2)	$(0.9, 0.999)$
Optimizer epsilon	1×10^{-8}
Learning rate	1×10^{-5}
Learning-rate schedule	Constant
Warmup ratio	0.03
Weight decay	0.0
Gradient clipping	1.0
Numerical precision	BF16
Random seed	42
Training epochs	3

Dataset-Specific Configurations. Table 5 records the data, input, and compute differences between the two benchmarks. WOD-E2E uses the full 400k-frame training set and 48 NVIDIA H200 GPUs. Each device processes 8 samples without gradient accumulation, giving an effective global batch size of 384. The inputs comprise RGB surround-camera observations, ego state, and a high-level navigation command; no LiDAR, radar, or HD map is used.

For nuScenes, the processed training and validation sets contain $15{,}394$ and $3{,}291$ samples, respectively. Training uses 8 NVIDIA H200 GPUs with per-device batch size of 8 and no gradient accumulation, giving an effective global batch size of 64. Each example contains three consecutive front-camera frames sampled at $t-1$ s, $t-0.5$ s, and t. This front-only input differs from the surround-camera WOD-E2E configuration and is therefore stated explicitly.

Table 5: Dataset-specific training and input configurations.

Setting	WOD-E2E	nuScenes
Training data	400k-frame training set	15,394-frame processed samples
Evaluation data	Official test set	3,291-frame validation set
Compute	48× NVIDIA H200	8× NVIDIA H200
Batch per device	8	8
Gradient accumulation	None	None
Effective global batch	384	64
Visual input	Front, front-left and front-right cameras	Three consecutive front-camera frames

HFLM and Decoding Hyperparameters. During training, the corruption time is sampled uniformly from $[0, 1]$, and the data-side endpoint retains Gaussian noise with $\gamma = 0.1$. At inference, the default VFD configuration uses 4 transport steps, freezes tokens above confidence threshold $\tau = 0.9$, and enables AR verification. The controlled efficiency experiment in Table 2 uses $N = 1$ to isolate the speed of a single drafting step; all other reported main configurations use the default $N = 4$.

C Experimental Results on nuScenes

Dataset and Protocol. We use nuScenes (Caesar et al., 2020) which contains $1{,}000$ urban driving scenes collected in Boston and Singapore, with the official $700/150/150$-scene train/validation/test split. Our processed split contains $15{,}394$ training samples and $3{,}291$ validation samples. Following the ST-P3 evaluation protocol used by OpenDriveVLA (Zhou et al., 2026), we report L2 trajectory error at 1, 2, and 3 seconds and the average across these horizons. The evaluation is performed on the validation split.

Training Configuration. We train for three epochs on 8 NVIDIA H200 GPUs. Each sample uses the three front-camera frames, while the optimization and decoder settings follow Appendix B. This experiment tests transfer to a different city distribution and sensor sampling protocol with a more restricted camera input than the WOD-E2E setup.

Table 6: L2 trajectory error (m) on the nuScenes validation set. Best in **bold**.

Method	1s ↓	2s ↓	3s ↓	Avg. ↓
UniAD (Hu et al., 2023)	0.44	0.67	0.96	0.69
VAD-Base (Jiang et al., 2023)	0.17	0.34	0.60	0.37
DriveVLM (Tian et al., 2025)	0.18	0.34	0.68	0.40
OmniDrive (Wang et al., 2025)	0.14	0.29	**0.55**	0.33
OpenDriveVLA (Zhou et al., 2026)	0.14	0.30	**0.55**	0.33
Hybrid-Flow Drive (Ours)	**0.11**	**0.24**	0.57	**0.31**

Results and Interpretation. Table 6 provides cross-dataset evidence under the above-mentioned protocol. Hybrid-Flow Drive obtains the lowest 1-s and 2-s errors among the listed methods, reducing them to 0.11 m and 0.24 m, respectively. Its average error is 0.31 m, compared with 0.33 m for the strongest listed prior averages.

The gain is concentrated at short horizons rather than uniformly distributed. At 3 seconds, the error of 0.57 m is 0.02 m above the best listed value of 0.55 m. Moreover, the compared systems may differ in sensors, input history, supervision, and training scale. The table should therefore be read as a benchmark comparison under the established ST-P3 metric rather than as a controlled component study. Together with the front-only input used here, these differences bound the scope of the cross-method conclusion.