UA2C: Uncertainty-Aware Adaptive Action Chunking for Offline-to-Online Decision-Making in Mixed Traffic

Hongki Kim [1] **Sangeun Park** [2] **Minhae Kwon** [2]

Abstract

Autonomous driving requires sequential decision-making under partial observations and heterogeneous interactions with human-driven vehicles. While action chunking provides a useful temporal abstraction by predicting multi-step future actions, fixed-length chunk execution can be unreliable when later actions are conditioned on stale observations. We propose **UA2C**, an uncertainty-aware adaptive action chunking framework for offline-to-online reinforcement learning (RL) in mixed traffic. UA2C first learns a flow-matching chunk policy from offline driving data and then refines the policy through online interaction. To account for behaviorally diverse surrounding vehicles, UA2C incorporates a driving-style inference module that augments the policy with local behavior context. During execution, UA2C estimates uncertainty from sampled action chunks and executes only a reliable prefix before replanning. Experiments show that UA2C improves offline reward and control smoothness over a one-step baseline, and further improves online performance over fixed chunk execution.

1. Introduction

Autonomous driving in mixed traffic requires decision-making under interactive and partially observable dynamics, where surrounding human-driven vehicles exhibit heterogeneous intentions, driving styles, and reactions [2, 5, 7, 12, 14]. Because surrounding vehicles may react differently to the same action depending on their driving styles, the consequence of a driving decision cannot be assessed from a single timestep alone [9, 10, 21, 27]. Driving strategies should therefore reason over temporally extended behavior rather than isolated one-step actions [20, 26].

Action chunking provides such a temporal abstraction by predicting a short sequence of future actions at each planning step, which improves behavioral consistency and extends the effective planning horizon [13, 29]. In interactive traffic, however, fixed-length execution creates a temporal commitment problem: executing more actions preserves coherent multi-step behavior but increases the risk of stale, interaction-mismatched tail actions, whereas replanning too early reduces chunking to myopic control [1, 16]. Thus, the execution length should be adapted online according to chunk reliability, rather than fixed a priori [17, 22, 29].

We propose **UA2C**, an uncertainty-aware **adaptive action chunking** framework for offline-to-online RL in mixed-traffic autonomous driving. UA2C learns temporally extended driving behaviors from offline data and refines the **flow-matching** chunk policy through online interaction [6, 19]. Instead of committing to a fixed-length action chunk, the policy adaptively executes a reliable prefix of each predicted chunk before replanning. To handle heterogeneous surrounding vehicles, UA2C **infers local driving styles** and incorporates them into both the actor and critic context. This design preserves coherent multi-step control while enabling reactive replanning under uncertain, diverse interactions.

2. Problem Formulation

2.1. Road Environment and Vehicles

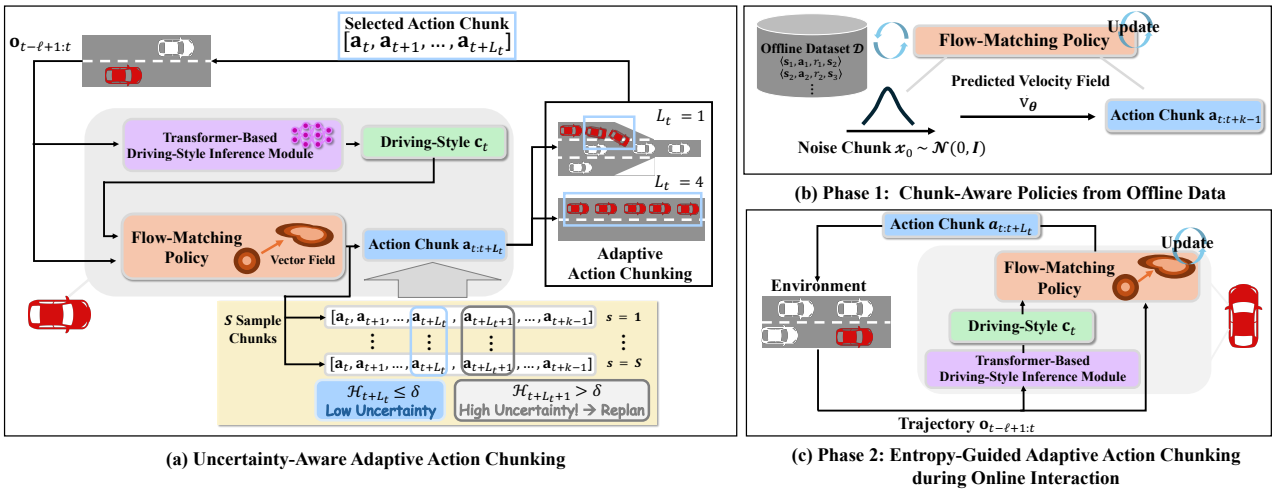
We consider a mixed-traffic environment with one autonomous vehicle and $N-1$ human-driven vehicles [3, 24]. To model a realistic environment, the autonomous vehicle partially observes H lanes within a distance of $2V$ [8, 11]. For each observable lane $h \in \{1, \dots, H\}$, the nearest vehicles ahead of and behind the autonomous vehicle are denoted by the *leader* l_h and *follower* f_h, respectively.

2.2. Partially Observable Markov Decision Process

We model the driving task as a partially observable Markov decision process, defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \Omega, \mathcal{R}, \gamma \rangle$,

[1]School of AI Software, Soongsil University, Republic of Korea [2]Department of Electrical and Computer Engineering, Sungkyunkwan University, Republic of Korea. Correspondence to: Minhae Kwon <minhae.kwon@skku.edu>.

Proceedings of the 43rd International Conference on Machine Learning Workshop on Decision-Making from Offline Datasets to Online Adaptation: Black-Box Optimization to Reinforcement Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).



(a) Uncertainty-Aware Adaptive Action Chunking

(b) Phase 1: Chunk-Aware Policies from Offline Data

(c) Phase 2: Entropy-Guided Adaptive Action Chunking during Online Interaction

Figure 1. Overview of the proposed UA2C framework. (a) UA2C: Uncertainty-Aware Adaptive Action Chunking, (b) Phase 1: Offline Pre-Training, and (c) Phase 2: Online Adaptation for Adaptive Action Chunking

wherein $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{O}$ denote a state space, an action space, and an observation space, respectively. The remaining elements consist of a transition probability $\mathcal{T}$, an observation probability Ω, a reward function $\mathcal{R}$, and a temporal discount factor $\gamma \in [0, 1)$.

Observation. The observation $\mathbf{o}_t \in \mathcal{O}$ of the autonomous vehicle is composed as follows.

$$\mathbf{o}_t = \left[v_{t,I}, p_{t,I}, \Delta\mathbf{v}_t^\top, \Delta\mathbf{p}_t^\top, \boldsymbol{\rho}_t^\top, \boldsymbol{\zeta}_t^\top\right]^\top \tag{1}$$

In (1), $v_{t,I}$ and $p_{t,I}$ denote the absolute speed and position of the autonomous vehicle, $\Delta\mathbf{v}_t = [\Delta v_{t,l_1}, \dots, \Delta v_{t,l_H}, \Delta v_{t,f_1}, \dots, \Delta v_{t,f_H}]^\top$ and $\Delta\mathbf{p}_t = [\Delta p_{t,l_1}, \dots, \Delta p_{t,l_H}, \Delta p_{t,f_1}, \dots, \Delta p_{t,f_H}]^\top$ denote the relative speed and position to leaders and followers, while $\boldsymbol{\rho}_t = [\rho_{t,1}, \dots, \rho_{t,H}]^\top$ and $\boldsymbol{\zeta}_t = [\zeta_{t,1}, \dots, \zeta_{t,H}]^\top$ denote the traffic density and the actual lane distance, respectively.

Action. The control action $\mathbf{a}_t \in \mathcal{A}$ at time step t is defined as $\mathbf{a}_t = [a_{t,\text{acc}},\ a_{t,\text{lc}}]$, where $a_{t,\text{acc}} \in [a_{\min}, a_{\max}]$ denotes the acceleration action, while $a_{t,\text{lc}} \in \{-1, 0, 1\}$ denotes the lane-change action for right lane change, lane keeping, and left lane change, respectively.

Reward. The reward at time step t, denoted by $r_t = R(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$, is defined as $R(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) = \sum_{\alpha=1}^{7} \eta_\alpha \mathcal{R}_{t,\alpha}$, where $\mathcal{R}_{t,\alpha}$ denotes the α-th reward component and η_α is its corresponding weighting coefficient. The reward function is designed to balance task efficiency, driving comfort, and safety through seven components: target-speed reward, safety-distance penalties with respect to leaders and followers, proactive merge completion, jerk penalty, unnecessary lane-change penalty, and collision penalty.

2.3. Actor–Critic Structure

We adopt an actor–critic framework for sequential decision-making under partial observations. The actor π_θ proposes an action sequence $\mathbf{a}_{t:t+k-1}$ conditioned on the observation, while the critic Q_ϕ evaluates the output actions by estimating its expected return.

3. Proposed Solution

We propose **UA2C**, an uncertainty-aware adaptive action chunking framework for offline-to-online RL in mixed-traffic autonomous driving. UA2C learns a **chunk-aware flow-matching policy** from offline driving data, refines it through online interaction, and executes only a reliable prefix of each predicted chunk before replanning. To capture heterogeneous interactions with surrounding vehicles, UA2C incorporates **driving-style inference** into the policy generation and value estimation, enabling behavior-aware adaptation to diverse traffic conditions, as illustrated in Figure 1.

3.1. Phase 1: Chunk-Aware Offline Pre-Training

Chunk-Aware Flow-Matching Actor. This phase pre-trains a chunk-aware policy that captures the distribution of temporally extended driving behaviors from offline data and uses critic-based guidance to favor high-value action chunks. During pre-training, the chunk length k is fixed. Let $h_t = (\mathbf{o}_{t-\ell+1:t}, \mathbf{a}_{t-\ell:t-1})$ denote the ℓ-step history context. The actor models a conditional distribution over future k-step action chunks $\mathbf{a}_{t:t+k-1} \sim \pi_\theta(\cdot \mid h_t)$.

To model the conditional chunk distribution, we sample a noise chunk $x_0 \sim \mathcal{N}(0, I)$ and an offline action chunk $x_1 = \mathbf{a}_{t:t+k-1}$ from the offline dataset. For a flow time

$\tau \sim \mathcal{U}[0,1]$, the interpolated chunk is defined as $x_\tau = (1-\tau)x_0 + \tau x_1$. The actor learns the conditional velocity field $v_\theta(x_\tau, \tau \mid h_t)$ by minimizing the following objective.

$$\mathcal{L}_{\text{actor}}(\theta) = \mathbb{E}\Big[\|v_\theta(x_\tau, \tau \mid h_t) - (x_1 - x_0)\|_2^2\Big] - \lambda\,\mathbb{E}\Big[\min_{j\in\{1,2\}} Q_{\phi_j}(\mathbf{o}_{t-\ell+1:t}, \hat{x}_1)\Big] \quad (2)$$

In (2), $\hat{x}_1 = x_\tau + (1-\tau)v_\theta(x_\tau, \tau \mid h_t)$ denotes the estimated terminal action chunk, and λ controls the strength of value guidance. The first term learns the transport direction from noise to data, while the second term encourages the actor to generate higher-value action chunks.

For value estimation, we use twin Transformer [23]-based chunk critics Q_{ϕ_1} and Q_{ϕ_2}. The critics are combined through a clipped double-Q estimate to mitigate overestimation bias [4], and are trained by minimizing the chunk-level Bellman regression loss as follows.

$$\mathcal{L}_{\text{critic}}(\phi) = \sum_{j=1}^{2} \mathbb{E}\left[\left(Q_{\phi_j}(\mathbf{o}_{t-\ell+1:t}, \mathbf{a}_{t:t+k-1}) - y_t\right)^2\right] \quad (3)$$

In (3), the target value y_t is defined as follows.

$$y_t = \sum_{u=0}^{k-1} \gamma^u r_{t+u} + \gamma^k \max_q \min_{j\in\{1,2\}} Q_{\bar{\phi}_j}\left(\mathbf{o}_{t+k-\ell+1:t+k}, \hat{\mathbf{a}}^{(q)}_{t+k:t+2k-1}\right) \quad (4)$$

In (4), $\{\hat{\mathbf{a}}^{(q)}_{t+k:t+2k-1}\}_{q=1}^{Q}$ denotes candidate future chunks sampled at h_{t+k}.

Driving-Style Inference Module. To account for heterogeneous local traffic behaviors, we introduce a driving-style inference module that predicts the behavior class of each observable surrounding vehicle from its recent partially observed trajectory. Let $m \in \{l_1, \ldots, l_H, f_1, \ldots, f_H\}$ denote a leader or follower slot. For each slot m, we construct a target-specific trajectory sequence $Z_m = [z^{(m)}_{t-\ell+1}, \ldots, z^{(m)}_t]$, where each token encodes relative motion features, target-relative attributes, and local traffic context. A Transformer encoder maps this sequence to a categorical behavior distribution $c^{(m)}_t = \text{softmax}(f_\psi(Z_m))$, where $c^{(m)}_t$ denotes the inferred driving-style distribution for slot m.

The module is trained as a standard multi-class sequence classifier using the cross-entropy objective as follows.

$$\mathcal{L}_{\text{style}} = -\sum_m \sum_{b=1}^{C} y_{m,b} \log c^{(m)}_{t,b} \quad (5)$$

In (5), C denotes the number of behavior classes, $y_{m,b}$ denotes the one-hot label for class b of slot m, and $c^{(m)}_{t,b}$ represents the predicted probability of that class. The resulting slot-wise distributions are concatenated into a driving-style context vector,

$$\mathbf{c}_t = \left[c^{(l_1)}_t, \ldots, c^{(l_H)}_t, c^{(f_1)}_t, \ldots, c^{(f_H)}_t\right], \quad (6)$$

which is appended to the control observation during online adaptation.

3.2. Phase 2: Online Adaptation with Entropy-Guided Adaptive Action Chunking

Although the policy predicts a full k-step chunk, later actions may become unreliable as they are conditioned on increasingly stale observations. Therefore, instead of always executing the full chunk, UA2C determines a reliable prefix and replans afterwards.

Specifically, given sampled future chunks $[\{\hat{\mathbf{A}}^{(s)}_t\}_{s=1}^{S}, \quad \hat{\mathbf{A}}^{(s)}_t = [\hat{\mathbf{a}}^{(s)}_t, \ldots, \hat{\mathbf{a}}^{(s)}_{t+k-1}],]$ we estimate the uncertainty at each chunk index using entropy. Let $\mathcal{H}_u$ denote the entropy of the sampled action distribution at the u-th position across candidate chunks. We then define the executable prefix length as follows.

$$L_t = \max\Big\{u \in \{1, \ldots, k\} \;\Big|\; \mathcal{H}_v \le \delta,\ \forall v \in \{1, \ldots, u\}\Big\} \quad (7)$$

In (7), δ denotes an entropy threshold. If $\mathcal{H}_1 > \delta$, we set $L_t = 1$ to ensure that at least one action is executed before replanning. The controller then executes the first L_t actions of the selected chunk and replans afterwards.

During online adaptation, the same adaptive execution rule is used for rollout collection. The critic target is computed from the discounted return of the executed prefix together with a bootstrap value from the replanning state after L_t steps as follows.

$$y_t = \sum_{u=0}^{L_t-1} \gamma^u r_{t+u} + \gamma^{L_t} \max_q \min_{j\in\{1,2\}} Q_{\bar{\phi}_j}\left(\mathbf{o}_{t+L_t-\ell+1:t+L_t}, \hat{\mathbf{A}}^{(q)}_{t+L_t}\right) \quad (8)$$

In (8), $\{\hat{\mathbf{A}}^{(q)}_{t+L_t}\}_{q=1}^{Q}$ denotes candidate future chunks sampled in the next planning state.

Thus, the target in (8) should be interpreted as a chunk-level value target under adaptive execution semantics, balancing the realized prefix return with a bootstrap estimate over the next replanning state [28].

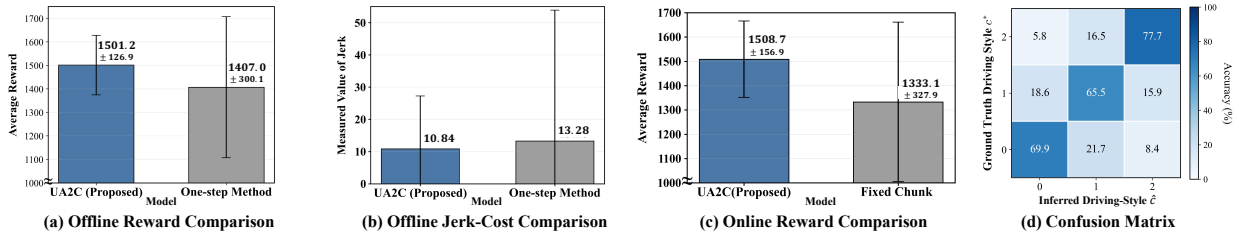


Figure 2. Performance comparisons. (a) Offline reward comparison with the one-step baseline, (b) Offline jerk-cost comparison with the one-step baseline, (c) Online reward comparison between fixed chunk execution and entropy-guided adaptive chunk execution, and (d) Confusion matrix for driving-style inference.

4. Simulation Results

To evaluate the proposed UA2C framework, we conduct simulations to assess three aspects: offline driving performance, online adaptation with **adaptive chunk** execution, and **driving-style inference** for surrounding vehicles.

4.1. Simulation Setup

We consider a mixed-traffic environment with one autonomous vehicle and 34 human-driven vehicles [3, 24]. The autonomous vehicle observes up to $V = 30\,\mathrm{m}$ ahead and behind, covering three lanes ($H = 3$) [15]. We implement the simulation using FLOW [25] with the SUMO traffic simulator [18]. To represent heterogeneous surrounding-vehicle behaviors, we consider three driving styles: conservative, which prioritizes safety and larger headway; neutral, which reflects moderate yielding and speed maintenance; and aggressive, which is characterized by shorter headway and more rapid acceleration or deceleration.

4.2. Offline Performance Comparison

We evaluate the effect of chunk-aware policy learning in the offline setting by comparing the proposed chunk policy with a one-step baseline. As shown in Figure 2(a), the chunk policy improves the average reward by 6.70% compared with the one-step baseline. This improvement suggests that chunk-level prediction provides a more informative temporal structure than isolated one-step actions. Figure 2(b) further shows that the chunk policy reduces cumulative jerk by 18.37%. Together, these results indicate that chunk-aware offline learning based on a **flow-matching policy** provides useful temporal abstraction for sequential driving control, leading to improved driving performance and smoother actions.

4.3. Online Adaptation Performance

We next evaluate the effect of adaptive chunk execution during online adaptation by comparing the proposed entropy-guided strategy with a fixed chunk execution baseline. While the fixed baseline always executes the full predicted chunk, the proposed method adaptively shortens the execution horizon according to the uncertainty of sampled future actions.

As shown in Figure 2(c), the proposed adaptive chunking method achieves a higher average reward than the fixed chunk baseline. In particular, the average reward improves by 13.17%, indicating that uncertainty-aware execution is more effective than naively committing to the full predicted chunk.

This result suggests that entropy-guided execution selects a more reliable execution horizon for predicted action chunks. In other words, **adaptive chunk execution** preserves the benefit of chunk-level prediction while avoiding unreliable tail actions under evolving traffic interactions.

4.4. Driving-Style Inference Performance

To evaluate the **driving-style inference module**, we report the confusion matrix on the test dataset in Figure 2(d). In the matrix, columns indicate the ground-truth driving style, while rows indicate the inferred driving style. The proposed module distinguishes the three driving styles overall. These results suggest that the inferred driving-style representation provides useful behavior context for adapting the policy to heterogeneous surrounding vehicles.

5. Conclusion

In this paper, we propose **UA2C**, a chunk-based offline-to-online driving framework that combines a flow-matching policy with a driving-style inference module in mixed traffic. We further introduce entropy-guided adaptive chunk execution, which selects a reliable prefix before replanning and enables stable online refinement. Simulation results show that UA2C improves offline driving performance and control smoothness over a one-step baseline, while adaptive execution further improves online performance over fixed chunk execution. More broadly, uncertainty-aware adaptive action chunking offers a general approach for balancing temporal coherence and reactive replanning in interactive sequential decision-making tasks.

Impact Statement

This paper advances offline-to-online RL methods for autonomous driving to improve decision-making under uncertainty and heterogeneous human-driver behaviors. By enabling adaptive action chunking, the proposed framework may contribute to more reliable driving policies in interactive traffic environments. We do not identify additional societal risks unique to this work beyond those generally associated with autonomous driving, including safety, accountability, and robustness under distribution shift.

References

[1] L. Ankile et al. From imitation to refinement–residual rl for precise visual assembly. In *Conference on Robot Learning (CoRL) Workshop on Mastering Robot Manipulation in a World of Abundant Data*, 2024.

[2] D. Chen, M. Hajidavalloo, Z. Li, K. Chen, Y. Wang, L. Jiang, and Y. Wang. Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic. *IEEE Transactions on Intelligent Transportation Systems*, 24(11):11623–11638, Nov. 2023.

[3] C. Eom, D. Lee, and M. Kwon. Selective imitation for efficient online reinforcement learning with pre-collected data. *ICT Express*, 10(6):1308–1314, Dec. 2024.

[4] S. Fujimoto et al. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning (ICML)*, 2018.

[5] C. Hubmann et al. Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles. In *IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[6] M. Kim, H. Park, and M. Kwon. Personalized split federated learning with early-exit: Pre-training and online learning against label shifts. *IEEE Internet of Things Journal*, 12(22), November 2025.

[7] D. Lee and M. Kwon. ADAS-RL: Safety learning approach for stable autonomous driving. *ICT Express*, 8(3):479–483, Sep. 2022.

[8] D. Lee et al. Episodic future thinking mechanism for multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[9] D. Lee and M. Kwon. Instant inverse modeling of stochastic driving behavior with deep reinforcement learning. *IEEE Transactions on Consumer Electronics*, 71(1):2152–2162, Feb. 2025. doi: 10.1109/TCE.2024.3464566.

[10] D. Lee and M. Kwon. Episodic future thinking with offline reinforcement learning for autonomous driving. *IEEE Internet of Things Journal*, 12(11):17012–17023, Jun. 2025.

[11] D. Lee and M. Kwon. Scenario-free autonomous driving with multi-task offline-to-online reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 26(9):13317–13330, Sep. 2025.

[12] J. Li, C. Yu, Z. Shen, Z. Su, and W. Ma. A survey on urban traffic control under mixed traffic environment with connected automated vehicles. *Transportation Research Part C: Emerging Technologies*, 154(9):104258, Sep. 2023. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2023.104258.

[13] Q. Li et al. Reinforcement learning with action chunking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[14] H. Liu et al. Reasoning multi-agent behavioral topology for interactive autonomous driving. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[15] J. Liu, P. Hang, X. Na, C. Huang, and J. Sun. Cooperative decision-making for CAVs at unsignalized intersections: A MARL approach with attention and hierarchical game priors. *IEEE Transactions on Intelligent Transportation Systems*, 26(1):443–456, Jan. 2025.

[16] S. Liu et al. BridgeDrive: Diffusion bridge policy for closed-loop trajectory planning in autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2026.

[17] Y. Liu et al. Bidirectional decoding: Improving action chunking via guided test-time sampling. In *International Conference on Learning Representations (ICLR)*, 2025.

[18] P. Lopez et al. Microscopic traffic simulation using SUMO. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[19] H. Park et al. Fed-ade: Adaptive learning rate for federated post-adaptation under distribution shift. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[20] S. Park et al. Multi^2: Hierarchical multi-agent decision-making with llm-based agents in interactive environments. In *International Conference on Machine Learning (ICML)*, 2026.

[21] H. Pei, S. Feng, Y. Zhang, and D. Yao. A cooperative driving strategy for merging at on-ramps based on dynamic programming. *IEEE Transactions on Vehicular Technology*, 68(12):11646–11656, Dec. 2019. doi: 10.1109/TVT.2019.2947192.

[22] J. So et al. Improving generative behavior cloning via self-guidance and adaptive chunking. In *Advances in neural information processing systems (NeurIPS)*, 2025.

[23] A. Vaswani et al. Attention is all you need. In *Advances in neural information processing systems (NeurIPS)*, 2017.

[24] J. Wang, Y. Zheng, Q. Xu, J. Wang, and K. Li. Controllability analysis and optimal control of mixed traffic flow with human-driven and autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(12):7445–7459, Dec. 2020.

[25] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. Bayen. FLOW: A modular learning framework for mixed autonomy traffic. *IEEE Transactions on Robotics*, 38(2):1270–1286, Apr. 2021.

[26] P. Wu et al. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[27] H. Xu, Y. Zhang, L. Li, and W. Li. Cooperative driving at unsignalized intersections using tree search. *IEEE Transactions on Intelligent Transportation Systems*, 21 (11):4563–4571, Nov. 2020. doi: 10.1109/TITS.2019.2940641.

[28] J. Yang et al. Actor-critic for continuous action chunks: A reinforcement learning framework for long-horizon robotic manipulation with sparse reward. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2026.

[29] T. Zhao et al. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.