

Autonomous Driving Economic Car-following Motion Strategy based on Adaptive Rollout Model-based Policy Optimization

Dong Hu, Chao Huang, *Senior Member, IEEE*, Jing Zhao, Yifan Zhao, *Senior Member, IEEE*,
and Jingda Wu *Member, IEEE*

Abstract—Reinforcement learning (RL) is a powerful framework with significant potential to enhance autonomous driving (AD) performance. However, its trial-and-error nature presents significant hurdles in terms of safety, efficiency, and stability. To address these challenges, we propose an adaptive rollout model-based policy optimization (AR-MBPO) algorithm tailored for car-following motion planning in autonomous electric vehicles (AEVs). The algorithm improves overall performance by incorporating a error-aware ensemble environment model and leveraging branched rollouts for efficient sample collection and policy optimization. A key innovation of AR-MBPO is an adaptive rollout mechanism that dynamically adjusts based on predictive accuracy, mitigating the impact of model inaccuracies. Additionally, energy efficiency is explicitly integrated into the optimization process to minimize energy consumption. We evaluate AR-MBPO through AEV car-following simulations, where it demonstrates superior performance, including rapid convergence, and reduced reliance on real-world interactions. The method simultaneously optimizes safety, traffic efficiency, and energy efficiency through dynamic distance adjustments, as evidenced by a 0% collision rate in testing scenarios and an 8.2% energy consumption reduction compared to non-energy-aware baselines. The results suggest potential applications in AD systems for improved safety and energy efficiency.

Index Terms—Model-based policy optimization, Autonomous electric vehicle, Adaptive rollout, Car-following

I. INTRODUCTION

WITH the growing integration of electric motors (EMs) and autonomous driving (AD), autonomous electric vehicles (AEVs) can optimize routes and speeds in real time, offering enhanced energy efficiency and driving comfort [1]. However, range anxiety and slow charging remain critical challenges [2], motivating the need for energy-efficient planning strategies.

In particular, longitudinal motion planning plays a crucial role in improving AEV energy efficiency. Traditional approaches often decouple energy optimization from motion

planning, relying on static driving cycles [3], and thus struggle in dynamic traffic scenarios. In contrast, reinforcement learning (RL) demonstrates superior adaptability in complex car-following tasks [4]. While RL offers strong real-time performance after training [5], its training efficiency is limited due to costly interactions and sparse data generation [6].

Model-based RL (MBRL) has recently gained attention for addressing this issue by simulating interactions between the environment and the agent using data-driven methods [7]. With an environment model, MBRL predicts state transitions based on the agent's actions, generating additional data to improve training efficiency. Methods like Dyna-style approaches [8], model predictive control (MPC) [9], value augmentation [10], and analytic gradient-based frameworks [11] are commonly used to build such models. However, accurately modeling complex driving environments remains challenging, as model inaccuracies can degrade performance [12]. Analytic gradient methods are particularly sensitive to modeling errors, while value augmentation and MPC-based methods often face importance sampling issues from inaccuracies [13]. These challenges make Dyna-style RL a promising alternative.

In Dyna-style approaches, rollout length is a key parameter, representing the number of simulation steps the environment model performs from the current state [14]. Existing studies often use heuristics or trial-and-error to manually set a fixed rollout length. Longer rollouts capture long-term task dependencies but quickly accumulate errors, while shorter rollouts reduce errors but lower training efficiency [15]. Finding an optimal rollout length and designing a dynamic adjustment strategy are critical challenges for Dyna-style RL-based AD strategies.

To address these gaps, we propose an adaptive rollout model-based policy optimization (AR-MBPO) strategy for car-following tasks in AEVs. Our approach dynamically adjusts rollout length based on real-time model accuracy, filtering unreliable simulated data. This improves training efficiency while ensuring robustness to model errors and dynamic conditions. The main contributions are as follows:

- We propose a novel AR-MBPO-based car-following strategy for AEVs, employing a error-aware ensemble model to predict state transitions and rewards, thereby improving training performance.
- We propose an adaptive rollout mechanism that dynamically adjusts rollout length based on environment model performance, enhancing adaptability.

The work has been supported by the Start-up Fund, PolyU (Grant No. P0039179). (*Corresponding author: Chao Huang.*)

Dong Hu, Chao Huang and Jingda Wu are with the Department of Industrial and Systems Engineering, the Hong Kong Polytechnic University, Hong Kong (E-mail: dong24.hu@connect.polyu.hk; hchao.huang@polyu.edu.hk; jingda.wu@polyu.edu.hk).

Jing Zhao is with the School of Mechanical Engineering and Automation, Northeastern University, Shenyang 110819, China, (e-mail: zhaoj@mail.neu.edu.cn;).

Yifan Zhao is with the School of Aerospace, Transport and Manufacturing, Cranfield University, MK43 0AL Bedfordshire, U.K. (e-mail: yifan.zhao@cranfield.ac.uk).

- The AR-MBPO strategy, which incorporates energy-saving objectives, is implemented for car-following tasks and validated across diverse testing scenarios.

Unlike studies that model powertrain and vehicle dynamics, our work focuses on high-level motion planning, enabling easy integration with different low-level models. To our knowledge, no studies have explored error-aware adaptive rollout mechanisms in Dyna-style MBRL for car-following motion planning.

The remainder of this paper is organized as follows: Section II reviews related research; Section III introduces the vehicle energy flow model; Section IV details the proposed approach; Section V describes its application in AD; Section VI presents and analyzes the results; and Section VII concludes the paper.

II. RELATED WORK

This section focuses on improving car-following in AD and optimizing motion with MBRL, both crucial for advancing autonomous transportation and tackling complex challenges.

A. Autonomous Driving Following Strategy

The car-following problem in AD involves real-time speed adjustments based on the leading vehicle's behavior to ensure safety, comfort and other indicators [16]. MPC, as an optimization-based method, use traffic data to plan speeds by solving complex optimization problems. For example, nonlinear MPC has been proposed to generate smooth, human-like trajectories in various scenarios [17]. However, MPC is computationally demanding and depends on accurate dynamic models, which can be difficult to obtain [18].

Among learning-based approaches, imitation learning is a common approach for car-following. Li et al. [19] proposed an interference-resistant strategy using behavior cloning to replicate expert driving behaviors, further optimized with self-supervised RL. Song et al. [20] introduced a hybrid approach combining RL for multi-objective cooperative optimization and supervised learning for human-like behavior imitation, achieving high-performance following control. Zhang et al. [21] developed an ACC-EMS strategy based on hierarchical RL, improving training speed and stability in simulations. Tang et al. [22] used two deep Q-networks (DQNs) for following control and hybrid powertrain energy management. Additionally, Wang et al. [4] demonstrated that multi-agent RL outperforms MPC in overall performance by coordinating the power system and following behavior.

However, RL faces significant challenges in car-following strategies due to the complexity of traffic scenarios. The inefficient training process of RL struggles to handle such complexity, requiring substantial data and time [23]. Therefore, improving RL's training quality is essential to address these issues and reduce trial-and-error reliance in real-world scenarios.

B. Model-based Reinforcement Learning

MBRL, a key branch of RL, uses interaction data to learn models of environmental dynamics, which are then employed

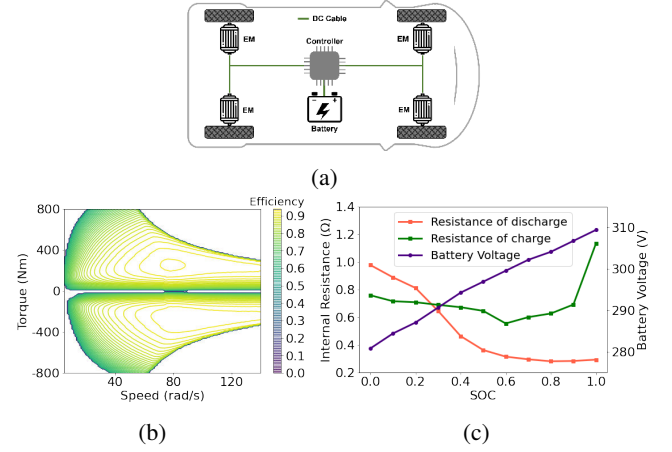


Fig. 1: (a) Electric vehicle structure. And its energy-flow modeling: (b) Efficiency map of the motors. (c) Internal resistance and voltage of the battery.

to generate additional training data. In robot control, RL has achieved significant progress. OpenAI [24] trained flexible robotic arms using RL and sim-to-real methods, surpassing human performance. Levine et al. [25] applied guided policy search for learning complex manipulation from raw camera data, while Fazeli et al. [26] enabled multi-modal reasoning by integrating visual and tactile data.

In AD, MBRL is valued for its sample efficiency. Lee et al. [27] proposed an ecological driving strategy based on Dyna-Q for velocity planning, considering terrain and following distance to improve energy efficiency. Wu et al. [28] introduced Dyna-PPO, which uses Gaussian processes for continuous decision-making in AVs and incorporates bidirectional interaction between world and policy models. Pan et al. [29] developed an imagination-based MBRL approach that constructs environment models to predict other vehicles' motion, addressing sparse rewards. However, MBRL faces challenges in modeling complex driving environments, as model inaccuracies can degrade performance, limiting its suitability in safety-critical tasks [30]. To mitigate this, Wu et al. [31] improved lane-changing decisions by assessing model prediction uncertainty, achieving higher learning efficiency than model-free RL (MFRL). Despite these advances, there is still little research on optimizing the MBRL rollout process by evaluating prediction differences in environment models, especially for longitudinal motion strategies in AVs, indicating an important area for further study.

III. VEHICLE ENERGY FLOW MODELING

To evaluate AEV energy efficiency in car-following, a vehicle energy flow model is constructed. The study considers a pure EV equipped with lithium-ion batteries (LIBs) and four in-wheel hub motors, which are centrally controlled to ensure equal power distribution (see Fig. 1(a)). The analysis focuses on drive energy consumption, excluding internal energy optimization among motors.

The backward vehicle model is used, requiring the vehicle

to first meet the total demand force F_{demand} [32]:

$$F_{\text{demand}} = mg \cos \theta^* f + \frac{1}{2} \rho A C_D u_r^2 + mg \sin \theta^* + \delta m a, \quad (1)$$

where m is vehicle mass, g is the gravitational acceleration, θ^* corresponds to the road incline, f is the rolling resistance coefficient, ρ is the air density, C_D is the drag coefficient, A refers to the frontal surface area, u_r indicates the relative speed, δ is the rotational inertia factor, and a is the longitudinal acceleration.

The electric motor (EM) is described using a quasi-static model, with efficiency η_{em} as a function (see Fig. 1(b)) of output torque T_{em} and rotational speed ω_{em} :

$$\begin{aligned} \eta_{\text{em}} &= f(\omega_{\text{em}}, T_{\text{em}}), \\ P_{\text{em}} &= \omega_{\text{em}} T_{\text{em}}, \end{aligned} \quad (2)$$

where P_{em} is motor output power. When computing the required drive force and mapping it to motor torque, the torque demand is always clipped within the physical limits of the EMs (-800 to 800 Nm). This ensures that all simulated actions and energy consumption calculations adhere to realistic actuator constraints.

The LIB uses an equivalent circuit model [33]:

$$\begin{cases} P_{\text{batt}}(t) = V_{\text{oc}}(t) - R_0 I^2(t) \\ I(t) = \frac{V_{\text{oc}}(t) - \sqrt{V_{\text{oc}}^2(t) - 4 \cdot R_0 \cdot P_{\text{batt}}(t)}}{2 \cdot R_0} \\ SOC(t) = \frac{Q_0 - \int_0^t I(t) dt}{Q_{\text{batt}}}, \end{cases} \quad (3)$$

where $P_{\text{batt}}(t)$ is the output power during charge-discharge cycles, $V_{\text{oc}}(t)$ is the open-circuit voltage, R_0 is the internal resistance, $I(t)$ is the current at time t , Q_0 is the initial battery charge capacity, and Q_{batt} is the nominal capacity. The relationship between internal resistance, open-circuit voltage, and state of charge (SOC) is shown in Fig. 1(c). And the main parameters are listed in Table I.

TABLE I: Parameters of the vehicle.

System	Parameters	Values
Vehicle	Curb weight m	1874 kg
	Frontal area A	2.6 m ²
	Air resistance coefficient C_D	0.23
	Rolling resistance coefficient f	0.01 kg
Motor	Maximum torque T_{max}	800 Nm
	Maximum speed ω_{max}	1350 rpm
Battery	Capacity C_{bat}	225 Ah
	Voltage V_{bat}	347.8 V

Remark 1: This study incorporates specific simplifications to mitigate model complexity. The longitudinal model assumes pure rolling, which may overlook tire slip under variable torque. Similarly, the quasi-static models for the EM and LIB ignore dynamic effects like transient torque control and charge/discharge processes. While these simplifications may limit accuracy in capturing certain dynamics, they are sufficient for evaluating the proposed motion strategy.

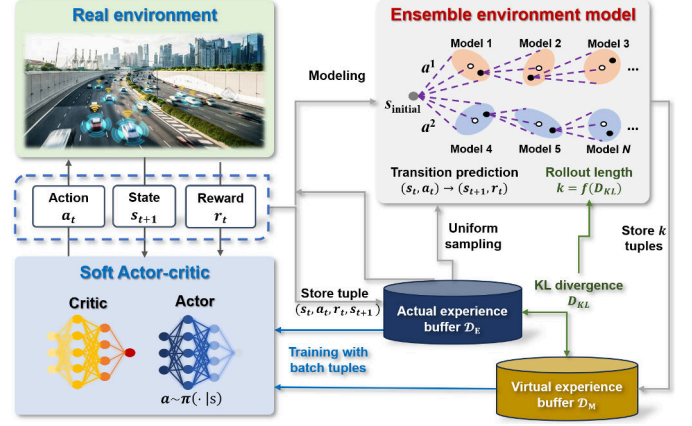


Fig. 2: The framework of the proposed AR-MBPO method.

IV. ADAPTIVE ROLLOUT MODEL-BASED POLICY OPTIMIZATION

A. Overview

This section presents the AR-MBPO framework, depicted in Fig. 2. The framework incorporates a soft actor-critic (SAC) agent that interacts with the real driving environment, storing collected interaction data in an experience buffer, referred to as $\mathcal{D}_E$. To improve learning efficiency and reduce the number of real-world interactions, the framework employs an adaptive branched rollout ensemble environment model. This model captures real transition dynamics and generates virtual experiences, represented by $\mathcal{D}_M$. Starting from actual states, the virtual model performs iterative rollouts using many sub-models with the same structure. Additionally, this ensemble-based approach dynamically adjusts rollout lengths to mitigate the challenges posed by unreliable predictions.

B. Soft Actor-Critic Algorithm

In driving tasks, the interaction between RL agents and the environment can be modeled as a Markov decision process (MDP), defined by tuples $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r)$. Here, $\mathcal{S}$ is the state space with state variables s_t , $\mathcal{A}$ is the action space with actions a_t , $\mathcal{T}(s_{t+1} | s_t, a_t)$ represents the environment's transition dynamics, and $r(s_t, a_t)$ is the reward function.

The SAC algorithm, an MFRL approach, learns a stochastic policy and excels in many standard environments. Its objective in the maximum entropy RL framework incorporates an entropy term, leading to the soft Bellman equation [34]:

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}} [V(s_{t+1})], \quad (4)$$

where γ is the discount factor, and the state-value function $V(s_t)$ is expressed as:

$$\begin{aligned} V(s_t) &= \mathbb{E}_{a_t \sim \pi} [Q(s_t, a_t) - \alpha \log \pi(a_t | s_t)] \\ &= \mathbb{E}_{a_t \sim \pi} [Q(s_t, a_t)] + H(\pi(\cdot | s_t)), \end{aligned} \quad (5)$$

where $H(\pi(\cdot | s_t))$ is the policy entropy, and α is the entropy regularization coefficient balancing exploration and exploitation.

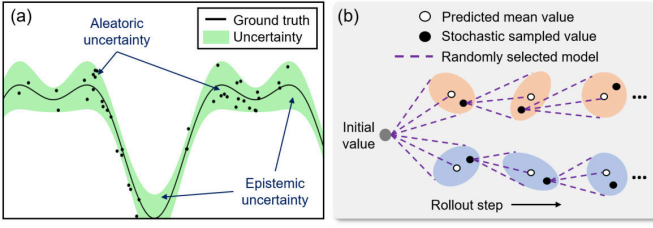


Fig. 3: (a) Uncertainty explanation and (b) sampling process of ensemble model.

SAC uses two action-value functions, Q_{ω_1} and Q_{ω_2} , with parameters ω_1 and ω_2 , and a policy π with parameters θ . The loss function for the action-value functions is defined as:

$$L_Q(\omega) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}} \left[\frac{1}{2} (Q_\omega(s_t, a_t) - (r_t + \gamma V_{\omega^-}(s_{t+1})))^2 \right], \quad (6)$$

where $\mathcal{D} = [\mathcal{D}_E, \mathcal{D}_M]$ represents the dataset collected by the policy, and Q_{ω^-} denotes the target network for stable updates. The loss function of the policy π is expressed as:

$$L_\pi(\theta) = \mathbb{E}_{s_t \sim \mathcal{D}, \xi_t \sim \mathcal{N}} \left[\alpha \log \pi_\theta(f_\theta(\xi_t; s_t) | s_t) - \min_{j=1,2} Q_{\omega_j}(s_t, f_\theta(\xi_t; s_t)) \right], \quad (7)$$

where $f_\theta(\xi_t; s_t)$ represents a reparameterization function that maps noise ξ_t to actions a_t under state s_t .

The entropy regularization coefficient α is adjusted automatically using:

$$L(\alpha) = \mathbb{E}_{s_t \sim \mathcal{D}, a_t \sim \pi(\cdot | s_t)} [-\alpha \log \pi(a_t | s_t) - \alpha \mathcal{H}_0], \quad (8)$$

where $\mathcal{H}_0$ is the target entropy value, which defines the degree of randomness in the policy.

C. Ensemble Environment Model

Under the MBRL framework, the environment model is established to enable virtual interactions between the agent and the environment model. The environment model, denoted as $\mathcal{M}$ with parameters ϕ , predicts the transition dynamics in driving scenarios. It takes as input the concatenation of the current state s_t and the action a_t , and outputs the predicted next state s_{t+1} along with the reward r_t . This mapping relationship is expressed as:

$$s_{t+1}, r_t \leftarrow \mathcal{M}_\phi(s_t, a_t). \quad (9)$$

However, the environment contains uncertainty, which can be categorized into two types [12]: the first, epistemic uncertainty, arises from incomplete knowledge or insufficient training data, such as unusual road conditions, or unpredictable behaviors of traffic participants; the second, aleatoric uncertainty, is caused by inherent randomness (Fig. 3(a)). To address these uncertainties, we adopt an ensemble learning approach inspired by probabilistic ensembles with trajectory sampling [35]. In this framework, candidate action sequences are sampled and evaluated using the environment model. The cross-entropy method is then employed to iteratively refine

the sampling distribution, making the search for high-reward actions more efficient and effective compared to random shooting. This approach captures epistemic uncertainty through model diversity and handles aleatoric uncertainty by modeling environment outputs as Gaussian distributions (Fig. 3(b)).

To address aleatoric uncertainty, we assume that the model outputs follow a Gaussian distribution $\mathcal{N}$, which can be described as:

$$\mathcal{M}_\phi \rightarrow \mu_\phi, \Sigma_\phi, \quad (10)$$

where μ_ϕ is the predicted mean, Σ_ϕ is the predicted covariance matrix, and both are parameterized by neural networks. The transition dynamics are then expressed as:

$$\mathcal{M}(s_t, a_t) = \mathcal{N}(\mu_\phi(s_t, a_t), \Sigma_\phi(s_t, a_t)). \quad (11)$$

To train the environment model, we use the negative log-likelihood loss function:

$$\mathcal{L}_\mathcal{M}(\phi) = \mathbb{E} \left[\frac{\log \Sigma_\phi}{2} + \frac{(d_t - \mu_\phi)^2}{2\Sigma_\phi} + c \right], \quad (12)$$

where d_t denotes the outputs generated by the model, which include the next step's state and reward, while c represents a constant value.

When the environment model encounters invisible states (e.g., situations it has not been trained on), increased epistemic uncertainty arises. This is directly related to inaccurate predictions. To capture epistemic uncertainty, we leverage an ensemble of N neural networks, where $N = 5$ in this study. Each neural network in the ensemble has an identical architecture but different parameter initializations. These networks are trained on randomly sampled subsets of real-world data and with varying data orderings. The ensemble is expressed as:

$$\mathcal{M}_{\text{ens}} = \mathcal{M}_\phi^1, \mathcal{M}_\phi^2, \dots, \mathcal{M}_\phi^N, \quad (13)$$

where $\mathcal{M}_\phi^i$ represents the i -th model in the ensemble. During prediction, the ensemble outputs a distribution over the next states. For trajectory sampling, the model selects one network from the ensemble for each prediction step, ensuring diversity and robustness. Unless otherwise specified, the symbol $\mathcal{M}$ and the environment model represent ensemble.

D. Monotonic Guarantee for Performance Improvement

MBPO typically aims to improve policy performance within a learned environment model. However, this does not necessarily guarantee improved performance in the real environment due to model inaccuracies and policy shifts. To ensure real environment safety and effectiveness, we derive a monotonic performance guarantee that constrains the disparity between the model environment and the real environment.

The expected return of a policy π in the real environment, denoted as $\eta[\pi]$, can be lower-bounded by its expected return in the model environment, $\hat{\eta}[\pi]$, penalized by a term C that accounts for model error and policy shift [36]:

$$\eta[\pi] \geq \hat{\eta}[\pi] - C, \quad (14)$$

where C is defined as:

$$C = 2r_{\max} \left[\frac{\gamma(\epsilon_{\mathcal{M}} + 2\epsilon_{\pi})}{(1-\gamma)^2} + \frac{2\epsilon_{\pi}}{(1-\gamma)} \right]. \quad (15)$$

Here, $\epsilon_{\pi} = \max_s D_{TV}(\pi \| \pi_D)$ quantifies the policy shift, which is defined as the total variation distance between the current policy π and the data collection policy π_D . Similarly, $\epsilon_{\mathcal{M}}$ measures the model generalization error, represented as the total variation distance between the real environment's transition dynamics $\mathcal{T}$ and the environment model's predictions $\mathcal{M}$:

$$\epsilon_{\mathcal{M}} = \max_t \mathbb{E}_{s \sim \pi_{D,t}} [D_{TV}(\mathcal{T}(s', r | s, a) \| \mathcal{M}(s', r | s, a))], \quad (16)$$

where $D_{TV}(\cdot \| \cdot)$ represents the total variation distance. In the this virtual model, the cumulative return can approximate the cumulative return of the real environment within a small error region C . However, C , as an indicator of model accuracy, has become a critical bottleneck for MBPO performance. In other words, reducing C throughout the training process is essential to further improve RL effectiveness.

E. Branched Rollout

1) *Theoretical Analysis*: In Eq.(14), if the model's generalization error is substantial, there may not exist a positive rollout length k that minimizes the rollout error for a given C , making it impossible to use the model. Therefore, the rollout length in the branched rollout must be finite to ensure that the model's generalization error remains manageable. If we use the current policy rather than the data collection policy $\pi_{D,t}$ from the current round to estimate the model error (denoted as ϵ'_m), we have:

$$\epsilon'_{\mathcal{M}} = \max_t \mathbb{E}_{s \sim \pi_t} [D_{TV}(\mathcal{T}(s', r | s, a) \| \mathcal{M}(s', r | s, a))], \quad (17)$$

which can be linearly approximated as:

$$\epsilon'_{\mathcal{M}} \approx \epsilon_{\mathcal{M}} + \epsilon_{\pi} \frac{d\epsilon'_{\mathcal{M}}}{d\epsilon_{\pi}}. \quad (18)$$

Combining the Eq.(15) and k -step branched rollout, we can obtain a new bound on the expected policy return:

$$\eta[\pi] \geq \eta^{\text{branch}}[\pi] - 2r_{\max} \left[\frac{\gamma^{k+1}\epsilon_{\pi}}{(1-\gamma)^2} + \frac{\gamma^k\epsilon_{\pi}}{(1-\gamma)} + \frac{k}{1-\gamma}\epsilon'_{\mathcal{M}} \right], \quad (19)$$

where η^{branch} represents the policy's expected return obtained using the branched rollout method. With the above formula, we can determine the theoretically optimal rollout length [36], which is:

$$\tilde{k}^* = \operatorname{argmin}_k \left[\frac{\gamma^{k+1}\epsilon_{\pi}}{(1-\gamma)^2} + \frac{\gamma^k\epsilon_{\pi}}{(1-\gamma)} + \frac{k}{1-\gamma}\epsilon'_{\mathcal{M}} \right]. \quad (20)$$

where, it can be observed that for $\gamma \in (0, 1)$, as the rollout length k increases, $\frac{\gamma^{k+1}\epsilon_{\pi}}{(1-\gamma)^2} + \frac{\gamma^k\epsilon_{\pi}}{(1-\gamma)}$ decreases while $\frac{k}{(1-\gamma)}\epsilon'_{\mathcal{M}}$ increases. Furthermore, if ϵ' is small enough, considering the relationship $\epsilon'_{\mathcal{M}} \approx \epsilon_{\mathcal{M}} + \epsilon_{\pi} \frac{d\epsilon'_{\mathcal{M}}}{d\epsilon_{\pi}}$, if $\frac{d\epsilon'_{\mathcal{M}}}{d\epsilon_{\pi}}$ is small enough, then the optimal rollout length becomes positive, and in this case, branched rollout is an effective method.

Although the theoretical rollout length $\tilde{k}^*$ can be derived, it is challenging to accurately estimate the model error $\epsilon_{\mathcal{M}}$

and policy error ϵ_{π} in practical applications, as they dynamically change during training. The high computational cost of evaluating error terms for all possible rollout lengths k in high-dimensional spaces, the inability of a globally optimal length to adapt to local variations in model errors across the state space, and the lack of robustness in theoretical results—due to their reliance on sensitive assumptions—can lead to suboptimal performance when error estimates are imprecise.

This study proposes two key principles. First, C exhibits a non-monotonic relationship with the number of iterations k . As the rollout length increases, C grows indefinitely because the discount factor γ is less than 1. Therefore, over-reliance on the virtual environment model should be avoided due to its inaccuracies. Second, model error $\epsilon_{\mathcal{M}}$ causes C to increase monotonically, leading to performance degradation. Thus, when the prediction error $\epsilon_{\mathcal{M}}$ is large, the rollout length k should be shortened, and conversely, when $\epsilon_{\mathcal{M}}$ is small, k can be extended [37]. Accordingly, adaptive derivation that does not rely on a fixed theoretical roll length becomes a better solution.

2) *Adaptive Rollout*: To address occasional inaccuracies in the environment model—especially with longer rollout lengths—we evaluate model accuracy and adjust rollout length accordingly. This is done by comparing samples from the real experience pool $\mathcal{D}_E = (s_t, a_t, s_{t+1}, r_t)$ and the model-generated pool $\mathcal{D}_M = (s_k, a_k, s_{k+1}, r_k)$, and measuring their distribution difference using Kullback-Leibler (KL) divergence.

Longer rollouts in MBPO speed up training but may introduce more errors. To balance efficiency and accuracy, the rollout length k is dynamically adjusted based on model bias rather than being fixed. The expected rollout length k^* is calculated as:

$$k^* = \lfloor \text{clip}(\tau / \max(D_{KL}[s_{t+1} \| s_{k+1}]_{\text{batch}}, D_{KL}[r_t \| r_k]_{\text{batch}}), 0, k_{\max}) \rfloor, \quad (21)$$

where $D_{KL}(\cdot \| \cdot)$ is KL divergence. τ is a weighting factor that scales the rollout length, $k_{\max} = 5$ is the maximum rollout length, $\text{clip}(\cdot)$ limits the result within $[0, k_{\max}]$, and $\lfloor \cdot \rfloor$ denotes rounding to the nearest integer. The expected rollout length is dynamically adjusted based on model accuracy, allowing the mechanism to balance efficiency and error propagation.

For varying rollout lengths, the theoretical analysis remains applicable. Eq.(14)-(20) explicitly quantify the impact of model errors $\epsilon_{\mathcal{M}}$ and policy shift ϵ_{π} on performance. These errors serve as the core basis for adjusting the rollout length, ensuring that $\epsilon_{\mathcal{M}}$ and ϵ_{π} remain controlled in real time. This ensures the monotonicity and stability of policy returns, making the adaptive mechanism a natural extension of the theoretical results.

Remark 2: AR-MBPO leverages an environment model to generate additional interaction data for training, allowing it to achieve higher learning efficiency than MFRL. However, interactions derived from the virtual model inevitably deviate from those in the real environment. As a result, MBPO theoretically cannot surpass the asymptotic performance ceiling of MFRL. In practice, the adaptive rollout mechanism

dynamically adjusts the rollout length according to model reliability, improving learning process while limiting error propagation.

Lumping all these components together, the final algorithm is summarized in Algorithm 1.

Algorithm 1 AR-MBPO

```

1: Initialize policy network  $\pi_\theta$ , value networks  $Q_\omega$ ;
2: Set the maximum episodes number  $N_{ep}$ ;
3: Initialize real experience buffer  $\mathcal{D}_E$  and virtual experience
   buffer  $\mathcal{D}_M$ ;
4: for episode = 1: $N_{ep}$  do
5:   Train environment ensemble  $\mathcal{M}_{ens}$  using data from
      $\mathcal{D}_E$ ;
6:   while not done do
7:     Execute  $a_t = \pi_\theta(s_t)$  in the real environment;
8:     Observe  $s_{t+1}, r_t$  and store  $(s_t, a_t, r_t, s_{t+1})$  in  $\mathcal{D}_E$ ;
9:     Determine the appropriate rollout length  $k^*$  based
       on  $\mathcal{D}_E$  and  $\mathcal{D}_M$ ;
10:    while  $k \leq k^*$  do
11:      Randomly select a sub-model  $i$  from the en-
        semble:  $\mathcal{M}_\phi^i \sim \mathcal{M}_{ens}$ ;
12:      Execute  $a_k = \pi_\theta(s_k)$  in  $\mathcal{M}_\phi^n$ ;
13:      Observe  $s_{k+1}, r_k$  and store  $(s_k, a_k, r_k, s_{k+1})$ 
        in  $\mathcal{D}_M$ ;
14:      Let  $k \leftarrow k + 1$ ;
15:    end while
16:    Update  $\pi_\theta, Q_\omega$  using data both  $\mathcal{D}_M$  and  $\mathcal{D}_E$ ;
17:    Real environment transition  $s_t \leftarrow s_{t+1}$ ;
18:  end while
19: end for

```

V. APPLICATION TO AUTONOMOUS DRIVING

A. Autonomous Driving Scenarios

This study utilizes vehicle trajectory data from the Next Generation Simulation (NGSIM) project. The dataset focuses on an area approximately 500 meters (1640 feet) in length, encompassing six highway lanes, including one designated for high-occupancy vehicles. It captures various traffic conditions, such as free-flowing, moderately congested, and fully congested states during peak periods. The dataset provides precise vehicle position information at a sampling rate of 10 Hz. To improve data accuracy, the reconstructed NGSIM I-80 dataset was employed. Car-following events were identified using a car-following filter based on the method outlined in [38], yielding 1341 car-following scenarios. Each scenario met the following requirements: (1) both the leading vehicle (LV) and following vehicle (FV) remained in the same lane, and (2) the event duration exceeded 15 seconds to ensure sufficient data for analysis.

B. Statement of Objectives

The proposed algorithm is designed to ensure that the AEV can move forward in a smooth and efficient manner while

preventing collisions with the LV and maximizing energy efficiency. Therefore, the reward function is defined as:

$$r = r_{\text{safety}} + r_{\text{efficiency}} + r_{\text{comfort}} + r_{\text{energy}} + P_c, \quad (22)$$

where $P_c = -100$ is a penalty applied when a collision occurs between the FV and the LV.

To evaluate safety, we use Time-to-collision (TTC), a widely accepted metric that represents the remaining time before two vehicles collide. TTC is calculated as:

$$TTC(t) = \frac{\Delta s(t)}{\Delta v(t)}, \quad (23)$$

where Δs is the distance between the FV and the LV, and Δv is their relative velocity. Considering that AEVs typically exhibit faster reaction times than human drivers, we set the lower limit for TTC to 3 seconds. The safety reward is then defined as:

$$r_{\text{safety}} = \begin{cases} \log\left(\frac{TTC}{3}\right), & 0 \leq TTC \leq 3, \\ 0, & \text{otherwise.} \end{cases} \quad (24)$$

The design of piecewise functions ensures that agents prioritize collision avoidance in critical situations and are not overly conservative in non critical situations.

Time headway (THW) is another important metric that measures the time it takes for the FV and LV to reach a specific point. Previous studies [39] fitted NGSIM experimental data to a log-normal distribution, showing that a THW of 1.26s corresponds to the maximum reward value, while excessively short or long THWs result in lower rewards. Additionally, the reward encourages higher speeds to a certain extent. The efficiency reward is defined as:

$$r_{\text{efficiency}} = \frac{1}{x\sigma'\sqrt{2\pi}} e^{\frac{-(\ln x - \mu')^2}{2\sigma'^2}} (\text{THW} \mid \mu', \sigma') + w_1 \cdot v_{\text{ego}}, \quad (25)$$

where x is the distribution variable of THW, $\mu' = 0.4226$ and $\sigma' = 0.4365$ are the mean and log-standard deviation of the distribution, and $w_1 = 10^{-3}$ is the weight of the speed reward. By adopting a log-normal distribution, the efficiency reward effectively models empirically observed behaviors from real-world traffic data, ensuring that the reward aligns with human driving norms.

Jerk, which measures the smoothness of acceleration changes. To penalize high Jerk values effectively, a quadratic function is used to create steep gradients for large Jerk changes, thereby encouraging smoother driving behavior. The comfort reward is represented as:

$$r_{\text{comfort}} = -\frac{\text{Jerk}^2}{Z}, \quad (26)$$

where $Z = 3600$ is a normalization parameter that scales the feature to the range $[0, 1]$.

To encourage energy-efficient driving, the real-time energy consumption is included in the reward function. A linear function is used for this reward component, as it provides a direct proportional relationship between energy usage and the penalty in the reward. This is defined as:

$$r_{\text{energy}} = -w_2 \cdot Q_{\text{elec}}, \quad (27)$$

where Q_{elec} represents the real-time energy consumption in kWh, and $w_2 = 5$. These weights were selected through empirical tuning to balance learning stability, driving efficiency, and energy consumption based on observed policy performance.

C. State and Action Space

In car-following scenarios, defining an appropriate state space requires accounting for both the ego vehicle's internal dynamics and the external environment. The internal dynamics include EM efficiency, which depends on its speed and torque—variables that can be represented by the ego vehicle's velocity and acceleration. The external environment is characterized by the LV state. Accordingly, the state space is defined by the ego vehicle's velocity v_{ego} , the relative distance Δs , and the relative velocity Δv with respect to the LV. The resulting state space is expressed as:

$$\mathcal{S} = \{v_{\text{ego}}, \Delta s, \Delta v\}. \quad (28)$$

For the action space, the acceleration of the FV is selected as the action variable, constrained within -3 to 3 m/s^2 as suggested by [39]. When mapping this action to the physical vehicle model, the corresponding wheel torque is computed and further clipped within the EM's torque limits (-800 to 800 Nm), consistent with the energy flow modeling. The action space is represented as:

$$\mathcal{A} = \{acc\}. \quad (29)$$

D. Experimental Setting

1) *Simulation Setup*: The 1341 NGSIM car-following scenarios were split into 80% for training and 20% for testing. During training, a random scenario was selected in each episode to create dynamic traffic conditions. For energy performance evaluation, the initial SOC for each vehicle was set to 0.92, with a time resolution of 0.1 seconds ensuring strategy reliability. To account for RL training randomness, three different random seeds were used. The policy and supporting scripts for communication, control, and evaluation were implemented in Python, with RL and deep learning components built using PyTorch. Simulations were run on a computer with an AMD R7-4800H CPU and Nvidia RTX2060 MQ GPU.

2) *Baseline Methods*: First, we employ several popular RL algorithms, including SAC [34], deep deterministic policy gradient (DDPG) [40], and proximal policy optimization (PPO) [41]. DQN from demonstration (DQNfD) [42] is a leading algorithm that utilizes demonstration data; and we adapt it by using SAC as the core to form a variant named SACfD. UA-MBRL [31] is an uncertainty-aware MBRL approach that improves efficiency and performance via ensemble models. SMBPPO [43] is a safe MBRL method that ensures constrained exploration using Lagrangian PPO and environment dynamics learning. Diffusion-QL [44] is an offline RL method that integrates Q-learning with diffusion models to enhance reward-based exploration.

Second, we evaluate the energy awareness of our proposed policy against four baselines. Baseline A uses AR-MBPO without energy-saving objectives to assess the effect of energy

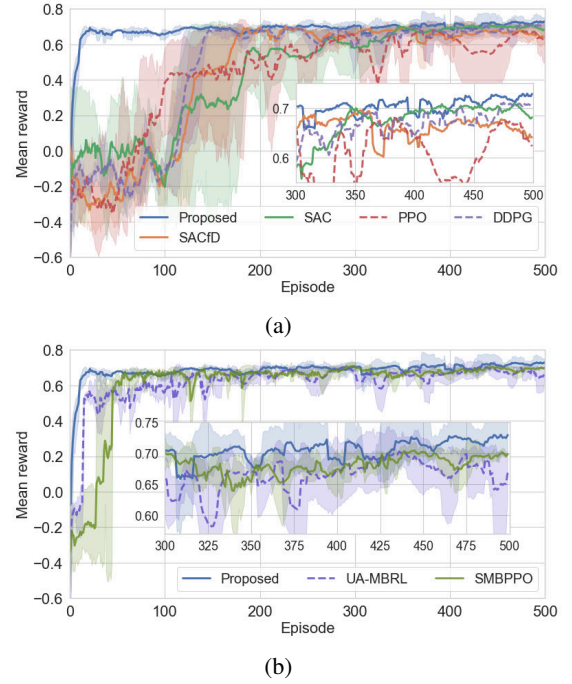


Fig. 4: Training reward curve: Comparison of the proposed scheme with (a) MFRL-based baselines and (b) MBRL-based baselines.

awareness. Baseline B applies a similar RL algorithm to our approach but does not use an environment model. Baseline C is similar to Baseline B but also omits energy-saving objectives. We further compare our policy with real human driving trajectories, as this provides a more effective benchmark for evaluating the the proposed strategy. Table II summarizes the main details of the four baseline policies.

TABLE II: Description of the adopted baseline strategies.

Name	Method	Highlighted features
Baseline A	AR-MBPO	With ablating the energy-related reward
Baseline B	Vanilla SAC	With the energy-related reward
Baseline C	Vanilla SAC	With ablating the energy-related reward
Human	Real data	The trajectory of human behavior

VI. RESULTS AND DISCUSSION

A. Training Results

Fig. 4(a) compares the training results of various RL algorithms over 500 episodes. The proposed approach achieves faster convergence and better asymptotic performance than other MFRL baselines. The average reward quickly rises to a high level, nearing convergence within 12 episodes and steadily improving afterward. AR-MBPO's efficiency stems from its use of model-assisted exploration, effective sampling, and stable policy optimization. Among the baselines, PPO performs poorly, with significant reward fluctuations due to varying driving scenarios across episodes, leading to instability in policy updates. SAC performs better asymptotically, as entropy regularization helps exploration and stabilizes training,

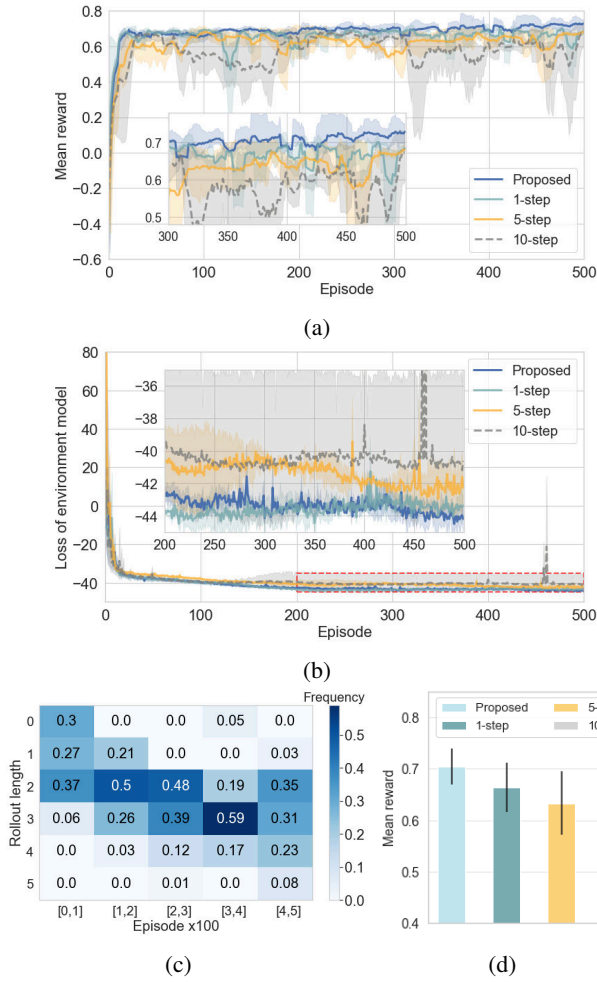


Fig. 5: Comparison of the proposed algorithm with the fixed length baseline method in terms of (a) training reward and (b) training loss of the environmental model. (c) Distribution of rollout lengths in AR-MBPO during training. (d) Average rewards over the final 300 episodes for the proposed algorithm and fixed-step baselines.

though the added randomness slows early progress. SACfD improves early convergence by incorporating demonstration data but risks deviating from the global optimum when over-relying on demonstrations in later stages. DDPG converges quickly with replay experience and target networks, but lacks regularization of entropy, resulting in mediocre asymptotic performance.

Fig. 4(b) compares the training rewards of MBRL baselines with the proposed approach. UA-MBRL achieves faster initial convergence due to its uncertainty-aware ensemble model and adaptive truncation but shows less stability and lower asymptotic performance in later stages. SMBPPO ensures safer training through constrained exploration, effectively reducing hazardous behaviors, but sacrifices convergence speed and sample efficiency.

B. Analysis of Rollout Length

Fig. 5 illustrates the comparison of the proposed algorithm with fixed-length baseline methods in terms of training reward,

TABLE III: Security performance of different methods.

Method	Train collision rate (%)	Test collision rate (%)
Proposed	0.50 ± 0.26	0.00 ± 0.00
SAC	13.63 ± 4.93	0.50 ± 0.43
PPO	14.03 ± 4.29	0.87 ± 0.22
DDPG	8.40 ± 2.19	0.00 ± 0.00
SACfD	10.77 ± 3.35	0.37 ± 0.37
UA-MBRL	1.27 ± 0.42	0.50 ± 0.57
SMBPPO	2.93 ± 1.01	0.00 ± 0.00
Diffusion-QL	- -	0.25 ± 0.21

The metric is represented as "mean $\pm$ standard deviation"

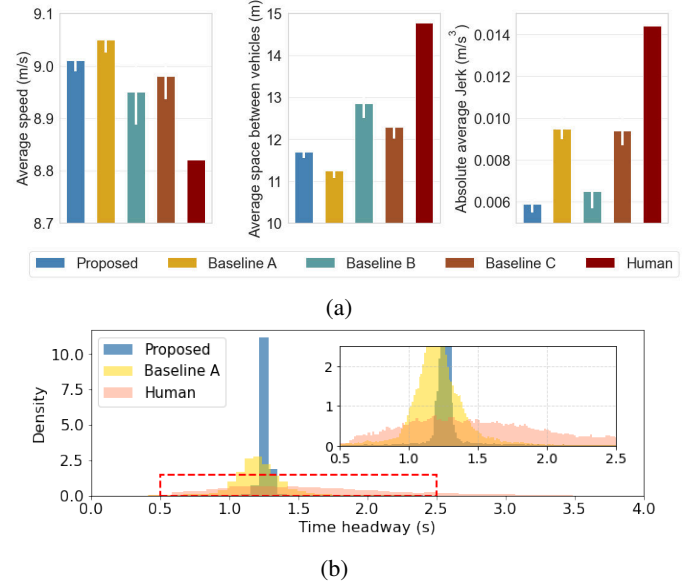


Fig. 6: Performance evaluation across different methods: (a) Comparison of average speed, spacing, and Jerk metrics; (b) Analysis of time headway distribution, a key indicator of traffic efficiency and safety.

rollout length adaptation, and environmental model loss. As shown in Fig. 5(a) and (d), the proposed method achieves the highest training reward and superior asymptotic performance, significantly outperforming fixed-length baselines. Specifically, the proposed method dynamically adjusts rollout lengths, mitigating the accumulation of model errors and achieving stable and high rewards even in later training stages. In contrast, fixed-length baselines, particularly those with longer rollout lengths (e.g., 10-step), experience a noticeable decline in performance due to compounding prediction errors.

Fig. 5(b) and (c) analyze the relationship between rollout lengths and environmental model performance. Fig. 5(b) compares the environment model's training loss across different methods. The proposed method achieves a training loss comparable to the 1-step baseline and remains the lowest throughout training, demonstrating high prediction accuracy and effective error control. In contrast, fixed-length baselines with longer rollouts (e.g., 5-step and 10-step) show much higher losses, indicating reduced prediction accuracy with increased rollout lengths. Fig. 5(c) shows the rollout length distribution in the proposed method during training. Early in

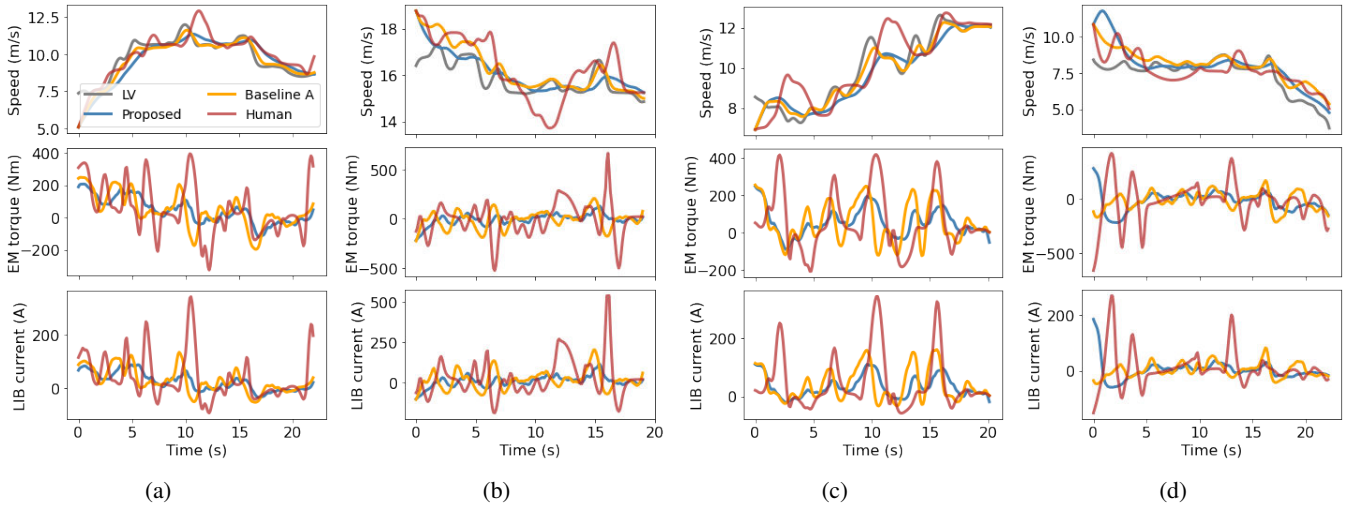


Fig. 7: Case study analyzing driving speed, EM torque, and LIB current trajectories: (a) Scenario 1; (b) Scenario 2; (c) Scenario 3; (d) Scenario 4.

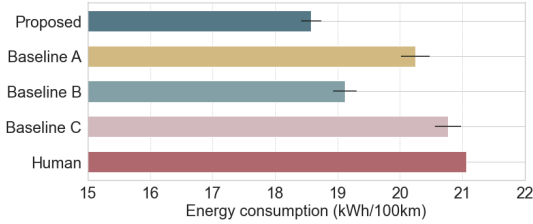


Fig. 8: Energy consumption results of different methods.

training, shorter rollout lengths dominate due to higher model errors and uncertainties. As training progresses and model accuracy improves, longer rollout lengths become more frequent, showcasing the adaptive mechanism’s ability to adjust rollout lengths based on model performance. These results demonstrate the effectiveness of the proposed adaptive mechanism in balancing model accuracy and sample efficiency, ensuring superior performance and stability.

C. Safety Performance Evaluation

Table III summarizes the average collision rates during 500 training episodes and 268 test scenarios under 3 random seeds. During training, the proposed method achieves a collision rate of 0.5%, significantly lower than SAC (13.6%), SACfD (10.8%), PPO (14.0%), and DDPG (8.4%). Among MBRL methods, UA-MBRL and SMBPPO achieve lower training collision rates of 1.27% and 2.93%, respectively, reflecting the high efficiency of their convergence. During testing, the proposed method, DDPG, and SMBPPO achieve 100% safety, with no collisions observed in any scenario. Diffusion-QL achieves a test collision rate of 0.25%, while UA-MBRL shows a slightly higher rate of 0.5%, similar to SAC. These results highlight the superior safety and reliability of the proposed strategy in car-following tasks, achieving the lowest collision rates during both training and testing phases, and outperforming both other MFRL and MBRL baselines.

D. Car-following Performance Evaluation

The evaluation environment consists of 268 car-following scenarios, where vehicles are controlled by well-trained policies for each method. Key performance metrics are analyzed both within and outside the vehicles, while their trajectory changes are closely monitored. As shown in Fig. 6(a), RL-based methods achieve approximate average vehicle speeds. Baselines A and C, which lack energy-related objectives, show a slight speed advantage due to their focus on traffic efficiency, often resulting in shorter following distances and more frequent acceleration. However, the speed difference compared to the proposed strategy is minimal, indicating that incorporating energy efficiency objectives into our RL approach does not compromise traffic efficiency. Notably, the proposed strategy achieves better average Jerk values, improving AEV comfort. In contrast, human-driven results fall significantly below all three performance metrics.

Additionally, as shown in Fig. 6(b), we evaluate the distribution of THW for different methods, which is a crucial indicator of both traffic efficiency and safety. The proposed strategy demonstrates a concentrated distribution around 1.0 to 1.5 seconds, balancing between safe following distances and efficient traffic flow. In contrast, Baseline A shows a narrower distribution at shorter THWs, suggesting a prioritization of traffic efficiency over safety. Human-driven results, however, exhibit a broader distribution with a tendency toward longer THWs, which compromises traffic efficiency. This analysis emphasizes the effective balance between security and driving efficiency achieved by the proposed strategy.

Fig. 7 illustrates four scenarios used to assess the performance of the proposed method in terms of vehicle speed, EM torque, and LIB current trajectories. For the speed trajectory, Baseline A, which does not consider energy consumption objectives, closely tracks the motion of the LV but demonstrates fewer speed oscillations due to its emphasis on driving comfort. In comparison, the proposed method integrates energy efficiency goals, leading to smoother accelerations

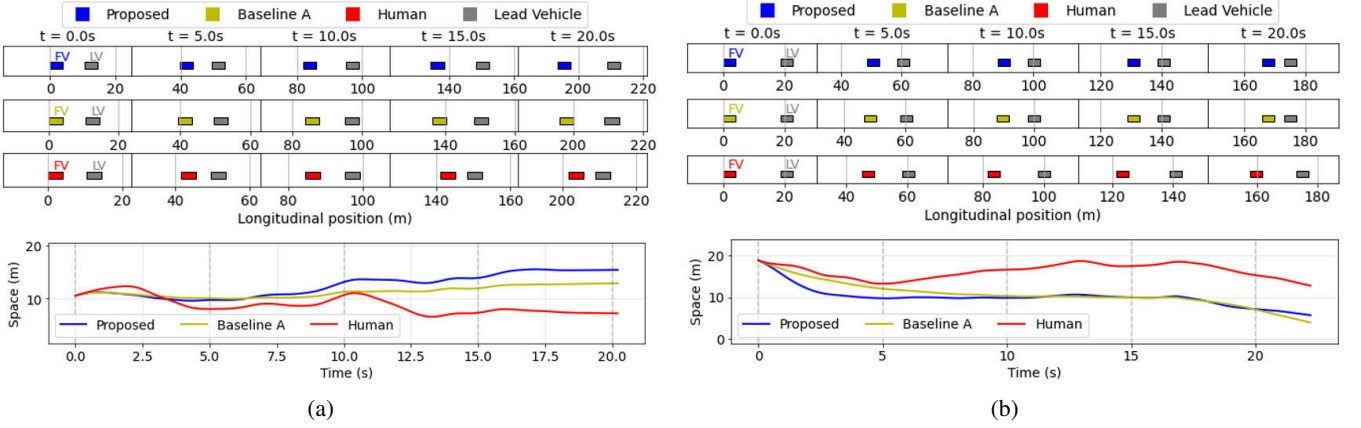


Fig. 9: Snapshots and spacing of car-following behavior using different methods in (a) Scenario 3 and (b) Scenario 4.

and reduced peak speeds. Meanwhile, human drivers display inconsistent speed variations caused by reaction delays, which can exacerbate traffic congestion.

E. Economic Performance Evaluation

Fig. 7 shows that EM torque and LIB current, which represent power and energy demand, further emphasize the advantages of the proposed strategy. The strategy achieves smaller torque amplitudes and lower current peaks, reflecting reduced energy consumption. Combined with lower Jerk values, it ensures a smoother and more stable driving process with smaller accelerations, enhancing energy efficiency. The reduced power demands result in lower EM torque, while smaller current fluctuations minimize resistive losses, improving LIB performance. It should be noted that, due to the high variability of torque profiles in Fig. 7, the pure rolling assumption may not strictly hold under all conditions. Nevertheless, these results are still sufficient to reveal the main trends and highlight the relative advantages of the proposed strategy in terms of energy consumption and control performance, as clarified in Section III.

Fig. 8 compares the energy consumption of various strategies, with the proposed method achieving the highest efficiency, consuming only 18.6 kWh per 100 kilometers. This represents the best performance among the evaluated strategies. Specifically, the proposed approach improves energy efficiency by 8.2% compared to the MBPO strategy without energy-awareness and by 11.5% relative to human driving. When compared to Baselines A and C, energy-aware strategies exhibit significantly better energy performance, highlighting the critical role of incorporating energy considerations into data-driven methods.

F. Analysis of Car-Following Process

Fig. 9 shows vehicle position snapshots at 5-second intervals and time-varying following distance curves of different methods, reflecting the position changes of the FV relative to the LV. According to Fig. 7, Scenario 3 corresponds to the acceleration process. Fig. 9(a) indicates that both RL methods can maintain relatively stable car-following distances.

The proposed method tends to maintain a larger following distance during acceleration, which enhances safety in high-speed scenarios and reduces the risk of rear-end collisions caused by sudden acceleration. In contrast, the human trajectory exhibits significant oscillations and tends to maintain a smaller following distance, increasing potential safety risks. Fig. 9(b) illustrates the car-following behavior in the deceleration scenario, showing that the proposed method tends to adjust in advance and maintain a smaller following distance during deceleration. This improves traffic efficiency in the deceleration phase while maintaining a more stable following distance in the later stage compared to Baseline A. On the other hand, the human trajectory consistently tends to maintain a larger following distance with significant oscillations, indicating relatively unstable behavior. Overall, maintaining a larger following distance during the acceleration phase helps reduce collision risks in high-speed conditions. Conversely, maintaining a smaller following distance during deceleration enhances road traffic efficiency, demonstrating the proposed method's ability to balance safety and efficiency across different scenarios.

TABLE IV: The test results of AR-MBPO reward component ablation.

Method	Collision (%)	Energy (kWh/100km)	Speed (m/s)	Jerk ($\times 10^{-3}$ m/s ³)
Full	0.00 $\pm$ 0.00	18.58 $\pm$ 0.16	9.01 $\pm$ 0.02	5.9 $\pm$ 0.4
$-r_{\text{safety}}$	6.97 $\pm$ 1.88	18.30 $\pm$ 0.22	9.11 $\pm$ 0.04	6.2 $\pm$ 0.5
$-r_{\text{efficiency}}$	0.00 $\pm$ 0.00	18.95 $\pm$ 0.19	8.78 $\pm$ 0.03	5.7 $\pm$ 0.4
$-r_{\text{comfort}}$	0.00 $\pm$ 0.00	18.89 $\pm$ 0.17	9.04 $\pm$ 0.02	11.4 $\pm$ 0.7
$-r_{\text{energy}}$	0.00 $\pm$ 0.00	20.25 $\pm$ 0.23	9.05 $\pm$ 0.02	8.9 $\pm$ 0.5

G. Ablation Study

To evaluate the contribution of each reward component, we performed an ablation study by independently removing the safety, efficiency, comfort, and energy terms from the reward function and comparing the resulting policy performance, as shown in Table IV. When all reward terms are included, the agent achieves a safe, energy-efficient, and smooth driving policy with high average speed. Removing the safety reward

results in a significant increase in collision rate, highlighting its critical role in collision avoidance. Excluding the efficiency term leads to a noticeable reduction in average speed, demonstrating that this term is essential for maintaining traffic flow. Omitting the comfort (Jerk) term causes a sharp rise in Jerk values with little effect on other metrics, indicating its specific role in improving ride comfort. Finally, removing the energy term increases overall energy consumption and slightly raises Jerk, while safety and speed remain mostly unchanged. These results confirm that each reward component is indispensable for balancing safety, efficiency, comfort, and energy consumption in the learned policy.

H. Calculation Efficiency Analysis

Table V reports the average inference time per inference step for different strategies. Although the proposed method is model-based during training, the environment model is not involved in the inference phase. As a result, its runtime efficiency is comparable to standard SAC.

TABLE V: The Average Inference Time for Different Strategies.

	Proposed	SAC	Diffusion-QL
Time/per step (ms)	2.41	2.28	4.47

VII. CONCLUSION

To enhance RL performance in AEVs, this study proposes the AR-MBPO method for multi-objective optimization. The main conclusions are as follows:

- 1) AR-MBPO achieves faster convergence than existing MFRL methods, stabilizing within approximately 12 episodes. This rapid learning substantially reduces safety risks inherent to RL exploration.
- 2) The adaptive rollout strategy consistently outperforms fixed rollout approaches, demonstrating that dynamic rollout adjustment effectively mitigates model inaccuracies and enhances robustness during training.
- 3) Testing shows that AR-MBPO balances safety, efficiency, comfort, and energy consumption. It matches non-energy-aware methods in safety and efficiency but surpasses them in comfort and economy, achieving an 8.2% improvement in energy efficiency over non-energy-aware AR-MBPO and an 11.5% gain compared to human driving.
- 4) The proposed method adapts well to diverse driving scenarios. In car-following tasks, it reduces collision risk by increasing following distance during acceleration and improves traffic efficiency by reducing gaps during deceleration, ensuring both safety and comfort.

Despite the promising results, this study has several limitations. First, the simplified modeling of tire-road interaction, motor, and LIB dynamics may limit generalization to real-world scenarios. Second, lateral dynamics and turning behaviors are not considered; incorporating advanced vehicle models could improve simulation fidelity and energy prediction accuracy. Finally, future work will focus on real-world validation

under noisy data and complex traffic, while integrating fallback strategies to enhance robustness and safety.

REFERENCES

- [1] P. Zhu, J. Hu, J. Li, F. Xiao, Z. Sun, and H. Peng, "Multistage prediction-based eco-driving control for connected and automated plug-in hybrid electric vehicles," *IEEE Transactions on Transportation Electrification*, vol. 10, no. 4, pp. 8030–8049, 2024.
- [2] W. Liu, T. Placke, and K. Chau, "Overview of batteries and battery management for electric vehicles," *Energy Reports*, vol. 8, pp. 4058–4084, 2022.
- [3] D. Hu, C. Huang, J. Wu, H. Wei, and D. Pi, "Enhancing data-driven energy management strategy via digital expert guidance for electrified vehicles," *Applied Energy*, vol. 381, p. 125138, 2025.
- [4] Y. Wang, Y. Wu, Y. Tang, Q. Li, and H. He, "Cooperative energy management and eco-driving of plug-in hybrid electric vehicle via multi-agent reinforcement learning," *Applied Energy*, vol. 332, p. 120563, 2023.
- [5] D. Hu, G. Zhou, J. Wu, and C. Huang, "Trust-calibrated human-in-the-loop reinforcement learning for safe and efficient autonomous navigation," *IEEE Internet of Things Journal*, pp. 1–1, 2025.
- [6] S. Feng, H. Sun, X. Yan, H. Zhu, Z. Zou, S. Shen, and H. X. Liu, "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, 2023.
- [7] D. Hu, L. Mo, J. Wu, and C. Huang, "'feariosity'-guided reinforcement learning for safe and efficient autonomous end-to-end navigation," *IEEE Robotics and Automation Letters*, vol. 10, no. 8, pp. 7723–7730, 2025.
- [8] B. Peng, X. Li, J. Gao, J. Liu, and K.-F. Wong, "Deep dyna-q: Integrating planning for task-completion dialogue policy learning," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2018, pp. 2182–2192.
- [9] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 7559–7566.
- [10] J. Buckman, D. Hafner, G. Tucker, E. Brevdo, and H. Lee, "Sample-efficient reinforcement learning with stochastic ensemble value expansion," *Advances in neural information processing systems*, vol. 31, 2018.
- [11] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.
- [12] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, 2017.
- [13] T. M. Moerland, J. Broekens, A. Plaat, C. M. Jonker, et al., "Model-based reinforcement learning: A survey," *Foundations and Trends in Machine Learning*, vol. 16, no. 1, pp. 1–118, 2023.
- [14] X. Wang, Z. Zhu, G. Huang, X. Chen, J. Zhu, and J. Lu, "Drivedreamer: Towards real-world-drive world models for autonomous driving," in *European Conference on Computer Vision*. Springer, 2024, pp. 55–72.
- [15] X. Zhou, M. Liu, E. Yurtsever, B. L. Zagar, W. Zimmer, H. Cao, and A. C. Knoll, "Vision language models in autonomous driving: A survey and outlook," *IEEE Transactions on Intelligent Vehicles*, pp. 1–20, 2024.
- [16] Q. Wang, F. Ju, H. Wang, Y. Qian, M. Zhu, W. Zhuang, and L. Wang, "Multiagent reinforcement learning for ecological car-following control in mixed traffic," *IEEE Transactions on Transportation Electrification*, vol. 10, no. 4, pp. 8671–8684, 2024.
- [17] C. Wei, E. Paschalidis, N. Merat, A. S. Crusat, F. Hajiseyedjavadi, and R. Romano, "Human-like decision making and motion control for smooth and natural car following," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 263–274, 2023.
- [18] B. Wingelaar, G. R. G. da Silva, and M. Lazar, "Model predictive eco-driving control for heavy-duty trucks using branch and bound optimization," *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [19] M. Li, Z. Li, S. Wang, and B. Wang, "Anti-disturbance self-supervised reinforcement learning for perturbed car-following system," *IEEE Transactions on Vehicular Technology*, 2023.
- [20] D. Song, B. Zhu, J. Zhao, J. Han, and Z. Chen, "Personalized car-following control based on a hybrid of reinforcement learning and supervised learning," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

- [21] H. Zhang, J. Peng, H. Dong, H. Tan, and F. Ding, "Hierarchical reinforcement learning based energy management strategy of plug-in hybrid electric vehicle for ecological car-following process," *Applied Energy*, vol. 333, p. 120599, 2023.
- [22] X. Tang, J. Chen, K. Yang, M. Toyoda, T. Liu, and X. Hu, "Visual detection and deep reinforcement learning-based car following and energy management for hybrid electric vehicles," *IEEE Transactions on Transportation Electrification*, vol. 8, no. 2, pp. 2501–2515, 2022.
- [23] Y. Lin, J. McPhee, and N. L. Azad, "Comparison of deep reinforcement learning and model predictive control for adaptive cruise control," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 221–231, 2021.
- [24] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, *et al.*, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.
- [25] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.
- [26] N. Fazeli, M. Oller, J. Wu, Z. Wu, J. B. Tenenbaum, and A. Rodriguez, "See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion," *Science Robotics*, vol. 4, no. 26, p. eaav3123, 2019.
- [27] H. Lee, K. Kim, N. Kim, and S. W. Cha, "Energy efficient speed planning of electric vehicles for car-following scenario using model-based reinforcement learning," *Applied Energy*, vol. 313, p. 118460, 2022.
- [28] G. Wu, W. Fang, J. Wang, P. Ge, J. Cao, Y. Ping, and P. Gou, "Dyna-ppo reinforcement learning with gaussian process for the continuous action decision-making in autonomous driving," *Applied Intelligence*, vol. 53, no. 13, pp. 16 893–16 907, 2023.
- [29] M. Pan, X. Zhu, Y. Zheng, Y. Wang, and X. Yang, "Model-based reinforcement learning with isolated imaginations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 2788–2803, 2023.
- [30] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [31] J. Wu, Z. Huang, and C. Lv, "Uncertainty-aware model-based reinforcement learning: Methodology and application in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 194–203, 2022.
- [32] S. Onori, L. Serrao, and G. Rizzoni, *Hybrid electric vehicles: Energy management strategies*. Springer, 2016, vol. 13.
- [33] Z. P. Cano, D. Banham, S. Ye, A. Hintennach, J. Lu, M. Fowler, and Z. Chen, "Batteries and fuel cells for emerging electric vehicle markets," *Nature energy*, vol. 3, no. 4, pp. 279–289, 2018.
- [34] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.
- [35] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," *Advances in neural information processing systems*, vol. 31, 2018.
- [36] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," *Advances in neural information processing systems*, vol. 32, 2019.
- [37] T.-L. Vuong and K. Tran, "Uncertainty-aware model-based policy optimization," *arXiv preprint arXiv:1906.10717*, 2019.
- [38] X. Wang, R. Jiang, L. Li, Y. Lin, X. Zheng, and F.-Y. Wang, "Capturing car-following behaviors by deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 910–920, 2017.
- [39] M. Zhu, Y. Wang, Z. Pu, J. Hu, X. Wang, and R. Ke, "Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving," *Transportation Research Part C: Emerging Technologies*, vol. 117, p. 102662, 2020.
- [40] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
- [41] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [42] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, *et al.*, "Deep q-learning from demonstrations," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.
- [43] A. K. Jayant and S. Bhatnagar, "Model-based safe deep reinforcement learning via a constrained proximal policy optimization algorithm," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 432–24 445, 2022.
- [44] Z. Wang, J. J. Hunt, and M. Zhou, "Diffusion policies as an expressive policy class for offline reinforcement learning," in *International Conference on Learning Representations (ICLR 2023)*, 2023.



Dong Hu (Graduate Student Member, IEEE) received the B.Eng. degree from Wuhan University of Technology, Wuhan, China, in 2020, and the M.Eng. degree from Tianjin University, Tianjin, China, in 2023. He is currently pursuing the Ph.D. degree with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong. His current research interests include reinforcement learning, autonomous vehicle navigation, and electrified vehicle optimization.



Chao Huang (Senior Member, IEEE) is a Research Assistant Professor at the Department of Industrial and System Engineering, The Hong Kong Polytechnic University (PolyU). Her research interests are human-machine collaboration, fault-tolerant control, mobile robots (EV, UAV), and path planning.

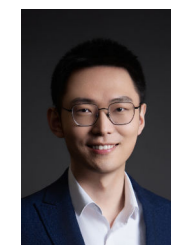
She serves as an Associate Editor for IEEE Transactions on Transportation Electrification, Engineering Applications of Artificial Intelligence, IEEE Transactions on Consumer Electronics, Control Engineering Practice, etc.



Jing Zhao received the Ph.D. degree in electromechanical engineering from the University of Macau, Macau, China. He is currently with the Automotive Engineering Laboratory, Department of Electromechanical Engineering, University of Macau. His research interests include vehicle dynamics and control, mechanism and machine theory, fluid mechanics, and finite element analysis.



Yifan Zhao (Senior Member, IEEE) received the Ph.D. degree in automatic control and system engineering from the University of Sheffield, Sheffield, U.K., in 2007. He is currently a Reader in data science with the Cranfield University, Cranfield, U.K. His research interests include machine learning, computer vision, signal processing, nondestructive testing, active thermography, and nonlinear system identification.



Jingda Wu (Member, IEEE) received the B.S. degree in vehicle engineering and the M.S. degree in mechanical engineering from Beijing Institute of Technology, Beijing, China, in 2016 and 2019, respectively, and the Ph.D. degree from the School of Mechanical and Aerospace Engineering, Nanyang Technology University, Singapore, in 2023.

His research interests include machine learning, autonomous vehicle planning, and electrified vehicle controls, where he has contributed to over 50 articles.



DSPACE

<https://dspace.org/>

Autonomous driving economic car-following motion strategy based on adaptive rollout model-based policy optimization

Hu, Dong; Huang, Chao; Zhao, Jing; Zhao, Yifan; Wu, Jingda

2025-10

Institute of Electrical and Electronics Engineers (IEEE)

<https://doi.org/10.1109/tte.2025.3590199>

Hu D, Huang C, Zhao J, et al., (2025) Autonomous driving economic car-following motion strategy based on adaptive rollout model-based policy optimization. IEEE Transactions on Transportation Electrification, Volume 11, Issue 5, October 2025, pp. 12416-12427