UNA-SAC: An Uncertainty-Aware Nonlinear Arbitration Method for Human-AI Shared Control

Shuyue Jiang, Yun-Bo Zhao, *Senior Member, IEEE,* Yu Kang, *Senior Member, IEEE,* Fei Xie, Yun-Sheng Zhao

***Abstract*—With the continuous development of artificial intelligence (AI), human-AI shared control has become an essential paradigm for achieving reliable collaboration, where the key challenge lies in efficiently arbitrating between human and AI policies. However, the inherent uncertainty of AI policies and their approximation errors often undermine the robustness and effectiveness of traditional linear arbitration. To address this issue, this paper proposes a nonlinear arbitration method based on the Soft Actor-Critic (SAC) framework, termed UNA-SAC. The method introduces a moment network to model AI policy uncertainty and incorporates a cognition-inspired mechanism to adjust the human policy, thereby constructing a distributional nonlinear arbitration form. Theoretical analysis demonstrates that the proposed method provides advantages in gradient optimization and effectively mitigates the cumulative effect of uncertainty-induced bias. Experimental results further validate its superiority in driving assistance scenarios: UNA-SAC achieves significant improvements in convergence speed, task success rate, robustness, and operational performance compared with linear arbitration and other baseline methods.**

***Impact Statement*—Human-AI shared control plays a vital role in safety-critical applications such as medical diagnosis, driving assistance, and unmanned aerial vehicle collaboration. However, the inherent uncertainty of AI policies, when fused with human policies, often leads to reduced reliability and robustness. Traditional linear arbitration methods are particularly susceptible to the amplification of estimation bias, thereby causing degradation in both training and execution performance. This paper proposes a nonlinear arbitration approach that can adaptively cope with AI policy uncertainty, offering a new solution for shared control. In driving assistance scenarios, the proposed method significantly improves convergence speed and execution performance, while demonstrating stronger stability and robustness, underscoring its broad potential for real-world applications.**

***Index Terms*—Shared Control, Reinforcement Learning, Nonlinear Arbitration, Uncertainty**

I. Introduction

With the rapid development of artificial intelligence (AI) technologies, human-AI shared control, by combining human cognitive advantages with the high-speed computation of AI, enables efficient and intelligent collaboration in complex environments [1]–[3]. This paradigm has been widely applied in key domains such as intelligent medical diagnosis [4], driving assistance systems [5], and unmanned aerial vehicle cooperative operations [6], significantly improving task efficiency and safety. However, many existing human—AI shared control methods still rely on simple linear arbitration, overlooking uncertainty and approximation errors in AI policies, which can lead to degraded assistance performance or even safety risks in complex traffic and highly dynamic flight scenarios [7], [8]. These limitations indicate the need for a more rigorous arbitration mechanism that explicitly accounts for uncertainty to enable reliable human—AI shared control.

In shared control, the arbitration mechanism serves as the core component for achieving effective collaboration, with its primary function being to fuse the policy outputs of humans and machine algorithms to generate the final control command [9]–[12]. When the machine algorithm is rule-based, arbitration is typically performed through linear weighting, where the control actions of humans and machines are combined according to weights [2]. Since in this case machine behavior is generated by predefined control logic, making the decision process predictable and stable, linear arbitration performs well in balancing autonomy and user intent [13]–[15].

In shared control with the introduction of AI algorithms, the uncertainty of AI policies weakens the effectiveness of linear arbitration. When the machine algorithm is AI-based, its policy is typically parameterized by deep learning models and optimized within supervised or reinforcement learning frameworks. Although such policies are adaptive, their inherent black-box nature inevitably introduces uncertainty [16]. More critically, existing uncertainty estimation methods themselves have limited accuracy [17]. If such estimates are directly employed in the weight computation of linear arbitration, estimation errors may cause the fusion ratio between human and AI to deviate from the desired value. These deviations exhibit a coupled amplification effect during the training iterations: incorrect weight allocation generates biased training data distributions, which in turn affect the update of the AI policy; the updated policy then produces new uncertainty estimation errors, further influencing subsequent weight allocation. Over long-term iterations, this coupled bias effect continuously amplifies policy deviation, leading to slower convergence, degraded policy quality, and reduced stability and safety in complex environments.

Therefore, existing linear arbitration methods for human-AI shared control exhibit inherent limitations: the bias in uncertainty estimation can be amplified during weight fusion and

This work was supported by the National Natural Science Foundation of China (No. 62173317). (Corresponding author: Yun-Bo Zhao.)

Shuyue Jiang, Fei Xie, and Yun-Sheng Zhao are with the Department of Automation, University of Science and Technology of China, Hefei 230026, China (e-mail: jiangsuy@mail.ustc.edu.cn; xiefei2021@mail.ustc.edu.cn; zys1030@mail.ustc.edu.cn).

Yun-Bo Zhao is with the Department of Automation, University of Science and Technology of China, Hefei 230026, China; the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088, China; and the Institute of Advanced Technology, University of Science and Technology of China, Hefei 230031, China (e-mail: ybzhao@ustc.edu.cn).

Yu Kang is with the School of Electrical Engineering and Automation, Hefei University of Technology, Hefei 230009, China; and the Department of Automation, University of Science and Technology of China, Hefei 230026, China (e-mail: kangduyu@ustc.edu.cn).

fed back into policy updates, resulting in distorted optimization and degraded performance.

However, existing studies are not yet systematic and largely rely on linear weighting, with little explicit modeling of the coupling between uncertainty estimation and arbitration. To address this issue, this paper proposes an Uncertainty-Aware Nonlinear Arbitration method, termed UNA-SAC, which is built upon the Soft Actor-Critic (SAC) [18] framework. The proposed method introduces a nonlinear arbitration mechanism into SAC, where human and AI policies are fused through the construction of a joint distribution, with uncertainty serving as a key regulatory factor to dynamically shape the arbitration distribution. The main contributions of this work are as follows:

1) **Nonlinear Arbitration Structure for Policy Fusion:** We design a nonlinear arbitration structure within the SAC framework. Specifically, we parameterize the AI policy as a Gaussian mixture and construct the arbitration policy via a normalized product of the human and AI policy distributions, thereby weakening the direct propagation of estimation bias during arbitration. In contrast to linear arbitration, this fusion mechanism exhibits more favorable gradient update properties in our theoretical analysis, alleviating the interference of uncertainty-induced bias in policy optimization.
2) **Cognition-Inspired Human Policy Adaptation under Uncertainty:** We propose a cognition-inspired human policy model in which AI policy uncertainty is quantified as the return variance. Specifically, a moment network estimates the second-order moment of returns, which is used to compute the variance. The variability of the human policy is then dynamically adjusted according to this uncertainty measure in a manner consistent with cognitive regulation. This mechanism enables the fusion process to adaptively respond to uncertainty fluctuations and reduces the impact of estimation bias on fusion accuracy.
3) **Comprehensive Evaluation in Driving Assistance Scenarios:** We conduct comprehensive experimental validation of the proposed method in driving assistance scenarios. The results demonstrate that it significantly outperforms linear arbitration and other baseline methods in terms of convergence speed and optimization performance.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III presents the problem formulation. Section IV introduces the preliminaries. Section V describes the proposed UNA-SAC method. Section VI provides the theoretical analysis. Section VII reports the experimental validation. Finally, Section VIII concludes the paper.

II. Related Work

Shared Control: Shared control methods are generally categorized into direct shared control and indirect shared control [19]. Direct shared control fuses human and machine outputs at the control-input level to generate the final control command. For example, [20] achieves shared stabilization under safety constraints, [21] realizes interactive optimization by online estimating the human feedback gain and cost weights, and [22] proposes adaptive weighting based on the angle between human and machine signals to enable smooth arbitration. In contrast, indirect shared control focuses on fusing human–machine information at the intention or policy level and indirectly influences the control input by evaluating human policies, allocating confidence, or adjusting the parameters of a unified controller. For instance, [23] performs policy-level fusion under a digital-twin framework, and [24] constructs a pilot–autopilot shared-control architecture based on the capacity for maneuver. Overall, direct shared control has a clear structure and is easier to implement and deploy in real time, but it is often difficult to explicitly express task-level intent and to consistently characterize policy uncertainty within input-level fusion; by comparison, indirect shared control can explicitly incorporate intent information and uncertainty measures at the policy or value-function level, making it more suitable for decision-level collaboration involving uncertainty-aware reasoning, albeit at the cost of more complex modeling and higher online computational overhead. Based on this comparison, our method adopts the indirect shared-control paradigm: it evaluates and regulates the human policy via the Human Policy module and constructs a nonlinear arbitration scheme at the policy-distribution level by incorporating AI policy uncertainty to generate the final shared-control command. Such distribution-level fusion allows uncertainty measures to enter the arbitration process explicitly and helps suppress the propagation of estimation bias during fusion, thereby improving robustness under uncertainty.

Linear Arbitration: Due to its simplicity and ease of implementation, this paradigm has been widely adopted for human-AI policy fusion in shared control [2]. Existing studies have applied linear weighting mechanisms across various tasks, such as policy blending to balance autonomy and user intent [13], customizable fusion parameters in robotic teleoperation [14], and shared linear quadratic regulator (sLQR) control for achieving minimal intervention based on reinforcement learning [15]. With the advancement of AI technologies, linear arbitration has also been employed in AI-driven shared control scenarios, such as dynamic adjustment mechanisms based on reachability value functions [25], and personalized control approaches with adaptive fusion weights [26]. However, in high-dimensional dynamic environments where AI policies exhibit significant uncertainty, such methods remain limited in the accuracy and robustness of weight allocation, making the fusion ratio prone to distortion due to estimation bias.

Nonlinear Arbitration: To overcome the limitations of linear arbitration in handling complex couplings within dynamic environments, several studies have explored nonlinear arbitration mechanisms. For example, [27] employs deep reinforcement learning to achieve shared autonomy by jointly embedding environmental observations and human inputs, selecting actions that are both high-valued and close to human preferences. [28] leverages a von Mises distribution to dynamically allocate control authority by identifying divergences between human and AI policies. [29] proposes a residual pol-

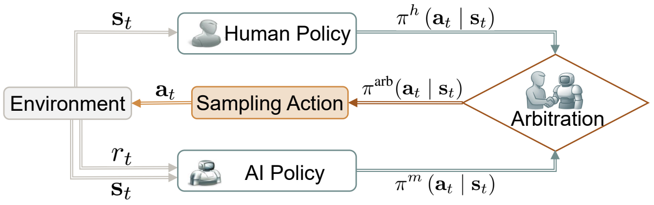
$\mathbf{s}_t$
Human Policy
$\pi^h(\mathbf{a}_t \mid \mathbf{s}_t)$
Environment
$\mathbf{a}_t$
Sampling Action
$\pi^{\text{arb}}(\mathbf{a}_t \mid \mathbf{s}_t)$
Arbitration
r_t
AI Policy
$\mathbf{s}_t$
$\pi^m(\mathbf{a}_t \mid \mathbf{s}_t)$

Fig. 1. Human-AI shared control framework.

icy learning method that enables minimal-intervention human-AI collaboration without relying on environment models or user goals. Although these methods enhance the flexibility and adaptability of arbitration, they do not account for the impact of AI policy uncertainty on system performance.

Uncertainty: For modeling the uncertainty of AI policies, existing methods can be broadly categorized into three types. The first is Bayesian neural networks based on variational inference [30], which often rely on independent assumptions such as factorized Gaussians to simplify computation, but fail to accurately characterize the true posterior. The second is stochastic sampling-based Monte Carlo dropout [31], whose estimates exhibit large fluctuations and insufficient stability. The third is distributional reinforcement learning with value functions [32], which is limited in its ability to capture tail risks. Although these methods provide uncertainty quantification for shared control, their reliability and accuracy remain limited in high-dimensional dynamic environments. When directly employed for linear arbitration, they can introduce systematic bias and lead to cumulative effects during training iterations, thereby degrading policy update quality and overall system performance.

III. Notation and Problem Formulation

In human-AI shared control, humans and AI collaborate to accomplish tasks, with dynamic allocation of control authority and policy fusion during the decision-making process. We model this process as a discrete-time Markov decision process (MDP) [33], denoted by a quadruple $(\mathcal{S}, \mathcal{A}, R, p)$, and the main mathematical notations are summarized in Table I. Specifically, $\mathcal{S}$ is the state space, and $\mathbf{s} \in \mathcal{S}$ represents the current system state; $\mathcal{A} \subset \mathbb{R}^d$ is the d-dimensional continuous action space, and $\mathbf{a} \in \mathcal{A}$ denotes an executable control action; $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, defined on a state–action pair $(\mathbf{s}, \mathbf{a})$ and returning an immediate reward r; $p : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the state transition function, where $p(\mathbf{s}' \mid \mathbf{s}, \mathbf{a})$ denotes the probability distribution of transitioning to the next state $\mathbf{s}'$ after taking action $\mathbf{a}$ in state $\mathbf{s}$.

At each time step t, conditioned on the current state $\mathbf{s}_t$, we model the human policy as the conditional distribution $\pi^h(\mathbf{a}_t \mid \mathbf{s}_t)$, which captures stochastic action selection and cognitive factors; in parallel, the AI decision module, parameterized by θ, defines the policy distribution $\pi^m_\theta(\mathbf{a}_t \mid \mathbf{s}_t)$. Instead of directly executing either one, the system employs an arbitration function $\beta : \mathcal{P}(\mathcal{A}) \times \mathcal{P}(\mathcal{A}) \to \mathcal{P}(\mathcal{A})$, to fuse the two into an arbitration policy

$$\pi^{\text{arb}}(\mathbf{a}_t \mid \mathbf{s}_t) = \beta\left(\pi^h(\mathbf{a}_t \mid \mathbf{s}_t),\, \pi^m_\theta(\mathbf{a}_t \mid \mathbf{s}_t)\right), \tag{1}$$

TABLE I
Summary of Main Notations

Symbol	Description
$\mathcal{S}$, $\mathbf{s}_t$	State space and state at time step t
$\mathcal{A}$, $\mathbf{a}_t$	Action space and action at time step t
r_t	Immediate reward at time step t
$\pi^m_\theta(\mathbf{a}_t \mid \mathbf{s}_t)$	AI policy with parameter θ
$\pi^h(\mathbf{a}_t \mid \mathbf{s}_t)$	Human policy under state $\mathbf{s}_t$
$\pi^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t)$	Nonlinear arbitration policy
$Q_\phi(\mathbf{s}_t, \mathbf{a}_t)$	Critic-estimated action-value function
R^E_t	Maximum-entropy cumulative return at time t
$M_\psi(\mathbf{s}_t, \mathbf{a}_t)$	Moment network output
$\mathrm{Var}^m(R^E_t \mid \mathbf{s}_t, \mathbf{a}^m_t)$	Return variance of the AI policy
$\mathrm{Var}^h(R^E_t \mid \mathbf{s}_t, \mathbf{a}^h_t)$	Return variance of the human policy
$\boldsymbol{\sigma}^2_h(\mathbf{s}_t)$	Variance vector of the human policy
K	Number of Gaussian mixture components
w_k	Weight of the k-th mixture component
$\pi^m_{\theta_k}(\mathbf{a}_t \mid \mathbf{s}_t)$	k-th Gaussian component of the AI policy
$\boldsymbol{\mu}_k(\mathbf{s}_t; \theta_k)$	Mean of the k-th AI Gaussian component
$\boldsymbol{\sigma}^2_k(\mathbf{s}_t; \theta_k)$	Variance of the k-th AI Gaussian component
$\pi^{\mathrm{NA}}_{\theta_k}(\mathbf{a}_t \mid \mathbf{s}_t)$	Fused k-th component after nonlinear arbitration
$Z(\mathbf{s}_t)$, $Z_k(\mathbf{s}_t)$	Normalization constants in arbitration
$\pi^{\mathrm{H}}_\varphi(\mathbf{a}_t \mid \mathbf{s}_t)$	Learned human model policy with parameter φ
$\mathcal{D}$, $\mathcal{D}_\mathcal{H}$	Replay buffer and human demonstration dataset

where π^{arb} is a valid probability distribution satisfying the normalization condition. The system then samples the final action $\mathbf{a}_t \sim \pi^{\text{arb}}(\cdot \mid \mathbf{s}_t)$ and applies it to the environment, which returns the immediate reward r_t and the next state $\mathbf{s}_{t+1}$. This process iterates throughout task execution, forming a human-AI collaborative interaction flow, as illustrated in Fig. 1.

IV. Preliminaries

In this study, the AI policy is modeled as a reinforcement learning agent based on SAC [18], denoted as $\pi^m_\theta(\mathbf{a}_t \mid \mathbf{s}_t)$. To clearly describe its optimization process, we first introduce the basic framework of SAC. SAC is built upon the maximum entropy reinforcement learning paradigm, with the optimization objective defined as:

$$\max_\theta \; \mathbb{E}_{\pi^m_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \left(r_t + \alpha\, \mathcal{H}(\pi^m_\theta(\cdot \mid \mathbf{s}_t))\right)\right], \tag{2}$$

where $\gamma \in (0, 1)$ is the discount factor, and $\alpha > 0$ is the temperature coefficient that controls the weight of the entropy regularization term. The policy entropy is defined as:

$$\mathcal{H}(\pi^m_\theta(\cdot \mid \mathbf{s}_t)) = \mathbb{E}_{\mathbf{a}_t \sim \pi^m_\theta}[-\log \pi^m_\theta(\mathbf{a}_t \mid \mathbf{s}_t)]. \tag{3}$$

To optimize the objective in Eq. (2), SAC models $\pi^m_\theta(\mathbf{a}_t \mid \mathbf{s}_t)$ with an actor network parameterized by θ, and introduces a critic network parameterized by ϕ to estimate the action-value function $Q_\phi(\mathbf{s}_t, \mathbf{a}_t)$. The loss function of the actor network is given by:

$$J(\pi^m_\theta) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D},\, \mathbf{a}_t \sim \pi^m_\theta}\left[\alpha \log \pi^m_\theta(\mathbf{a}_t \mid \mathbf{s}_t) - Q_\phi(\mathbf{s}_t, \mathbf{a}_t)\right], \tag{4}$$

where $\mathcal{D}$ denotes the replay buffer. The corresponding gradient form is:

$$\begin{aligned}\nabla_\theta J(\pi^m_\theta) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D},\, \mathbf{a}_t \sim \pi^m_\theta}\Big[\nabla_\theta\big(&\alpha \log \pi^m_\theta(\mathbf{a}_t \mid \mathbf{s}_t) \\ &- Q_\phi(\mathbf{s}_t, \mathbf{a}_t)\big)\Big].\end{aligned} \tag{5}$$

Based on the reparameterization trick,

$$\mathbf{a}_t = f_\theta(\boldsymbol{\epsilon}_t; \mathbf{s}_t), \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{6}$$

At this point, Eq. (5) can be rewritten as:

$$\begin{aligned}\nabla_\theta J(\pi_\theta^m) = \mathbb{E}_{\mathbf{s}_t,\,\boldsymbol{\epsilon}_t}\Big[&\alpha\,\nabla_\theta \log \pi_\theta^m(\mathbf{a}_t \mid \mathbf{s}_t) \\ &+ \big(\alpha\nabla_{\mathbf{a}} \log \pi_\theta^m(\mathbf{a} \mid \mathbf{s}_t) - \nabla_{\mathbf{a}} Q_\phi(\mathbf{s}_t, \mathbf{a})\big)\big|_{\mathbf{a}=\mathbf{a}_t} \\ &\times \nabla_\theta f_\theta(\boldsymbol{\epsilon}_t; \mathbf{s}_t)\Big].\end{aligned} \tag{7}$$

To improve the stability of value estimation, SAC introduces a double-critic architecture with Q_{ϕ_1} and Q_{ϕ_2}, and takes their minimum to mitigate overestimation bias:

$$Q_\phi(\mathbf{s}_t, \mathbf{a}_t) := \min(Q_{\phi_1}(\mathbf{s}_t, \mathbf{a}_t),\, Q_{\phi_2}(\mathbf{s}_t, \mathbf{a}_t)). \tag{8}$$

Meanwhile, target critic networks $Q_{\bar{\phi}_1}$ and $Q_{\bar{\phi}_2}$ are introduced, with their parameters updated through soft updates:

$$\bar{\phi}_i \leftarrow \tau\phi_i + (1-\tau)\bar{\phi}_i, \quad i = 1, 2, \tag{9}$$

where $\tau \in (0, 1)$ is the soft update coefficient. The target Q-value is defined as:

$$Q_{\bar{\phi}}(\mathbf{s}_t, \mathbf{a}_t) := \min(Q_{\bar{\phi}_1}(\mathbf{s}_t, \mathbf{a}_t),\, Q_{\bar{\phi}_2}(\mathbf{s}_t, \mathbf{a}_t)). \tag{10}$$

The constructed TD target is defined as:

$$y_t = r_t + \gamma\, \mathbb{E}_{\mathbf{a}_{t+1}\sim\pi_\theta^m}\left[Q_{\bar{\phi}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) - \alpha \log \pi_\theta^m(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1})\right], \tag{11}$$

Accordingly, the mean squared error loss of the critic network is defined as:

$$\mathcal{L}^Q(\phi_i) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})\sim\mathcal{D}}\left[\left(Q_{\phi_i}(\mathbf{s}_t, \mathbf{a}_t) - y_t\right)^2\right], \quad i = 1, 2. \tag{12}$$

Within the above SAC framework, the AI policy $\pi_\theta^m(\mathbf{a}_t \mid \mathbf{s}_t)$ is modeled by the actor network and continuously sampled and updated with parameters θ and ϕ during interactions with the environment, so as to optimize the maximum entropy objective and improve the expected return. Building upon this gradient update framework, the subsequent nonlinear arbitration mechanism incorporates human policy information and uncertainty fusion, thereby forming an arbitration policy optimization method for shared control.

V. Method

The proposed UNA-SAC framework integrates return uncertainty modeling with a nonlinear arbitration mechanism to achieve dynamic and adaptive human-AI collaboration. As illustrated in Fig. 2, the overall framework consists of five functional modules and forms a closed-loop optimization process through interaction with the environment and the replay buffer.

For the AI policy, the actor network parameterizes the policy $\pi_\theta^m(\mathbf{a}_t \mid \mathbf{s}_t)$ and is optimized based on the maximum entropy objective, while the critic network evaluates the value of state–action pairs, with its estimates used both to guide actor updates and to support uncertainty fusion. To quantify the return uncertainty of the AI policy, a moment network models the higher-order moments of the return distribution and outputs the estimate $M_\psi(\mathbf{s}_t, \mathbf{a}_t)$, providing an uncertainty measure for arbitration. This measure is further fed back to the human policy module, enabling it to dynamically adjust its action distribution according to the confidence of the AI policy and cognitive feedback.

The nonlinear arbitration module receives the distributions from the AI policy $\pi_\theta^m(\mathbf{s}_t, \mathbf{a}_t)$ and the human policy $\pi^h(\mathbf{s}_t, \mathbf{a}_t)$, and fuses them through an arbitration function to generate the final arbitration policy. The arbitration policy $\pi^{\mathrm{NA}}(\mathbf{s}_t, \mathbf{a}_t)$ is then used to interact with the environment to produce the execution action $\mathbf{a}_t$, while the interaction data $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$ are stored in the replay buffer for the joint update of all network modules.

A. Uncertainty Estimation

To quantify the execution uncertainty of the AI policy under the SAC framework, we introduce a modeling method based on the second moment of the maximum entropy return. Inspired by [34], we design an independent moment network $M_\psi(\mathbf{s}_t, \mathbf{a}_t)$ to approximate the conditional second moment of the maximum entropy cumulative return given state $\mathbf{s}_t$ and action $\mathbf{a}_t$. The maximum entropy cumulative return is defined as:

$$R_t^E := \sum_{l=0}^{\infty} \gamma^l \left(r_{t+l} + \alpha\, \mathcal{H}(\pi_\theta^m(\cdot \mid \mathbf{s}_{t+l}))\right), \tag{13}$$

The objective of the moment network is to compute

$$\mathbb{E}_{\pi_\theta^m}\left[(R_t^E)^2 \mid \mathbf{s}, \mathbf{a}\right]. \tag{14}$$

Correspondingly, the critic network estimates the first-order expectation of the maximum entropy return:

$$Q_\phi(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\pi_\theta^m}\left[R_t^E \mid \mathbf{s}, \mathbf{a}\right]. \tag{15}$$

Accordingly, the return variance can be expressed as

$$\mathrm{Var}_{\pi_\theta^m}(R_t^E \mid \mathbf{s}, \mathbf{a}) = M_\psi(\mathbf{s}, \mathbf{a}) - \left(Q_\phi(\mathbf{s}, \mathbf{a})\right)^2, \tag{16}$$

and serves as a quantitative measure of the uncertainty in policy outputs.

To train the moment network, we consider the squared form of R_t^E:

$$(R_t^E)^2 = \left(r_t + \gamma R_{t+1}^E\right)^2 = r_t^2 + 2\gamma r_t\, R_{t+1}^E + \gamma^2 (R_{t+1}^E)^2. \tag{17}$$

By taking the conditional expectation on both sides and sampling the next action according to the policy π_θ^m, we obtain:

$$\begin{aligned}M'(\mathbf{s}_t, \mathbf{a}_t) = r_t^2 &+ 2\gamma r_t\, \mathbb{E}_{\mathbf{a}_{t+1}\sim\pi_\theta^m}\left[Q_\phi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})\right] \\ &+ \gamma^2\, \mathbb{E}_{\mathbf{a}_{t+1}\sim\pi_\theta^m}\left[M_\psi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})\right].\end{aligned} \tag{18}$$

The mean squared error loss of the moment network is defined as:

$$\mathcal{L}^M(\psi) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})\sim\mathcal{D}}\left[\left(M_\psi(\mathbf{s}_t, \mathbf{a}_t) - M'(\mathbf{s}_t, \mathbf{a}_t)\right)^2\right]. \tag{19}$$

Since the update of the moment network is independent of the policy gradient optimization process, it can be embedded into the SAC framework as a separate module running in parallel with policy updates, thereby enabling the modeling of policy execution uncertainty.

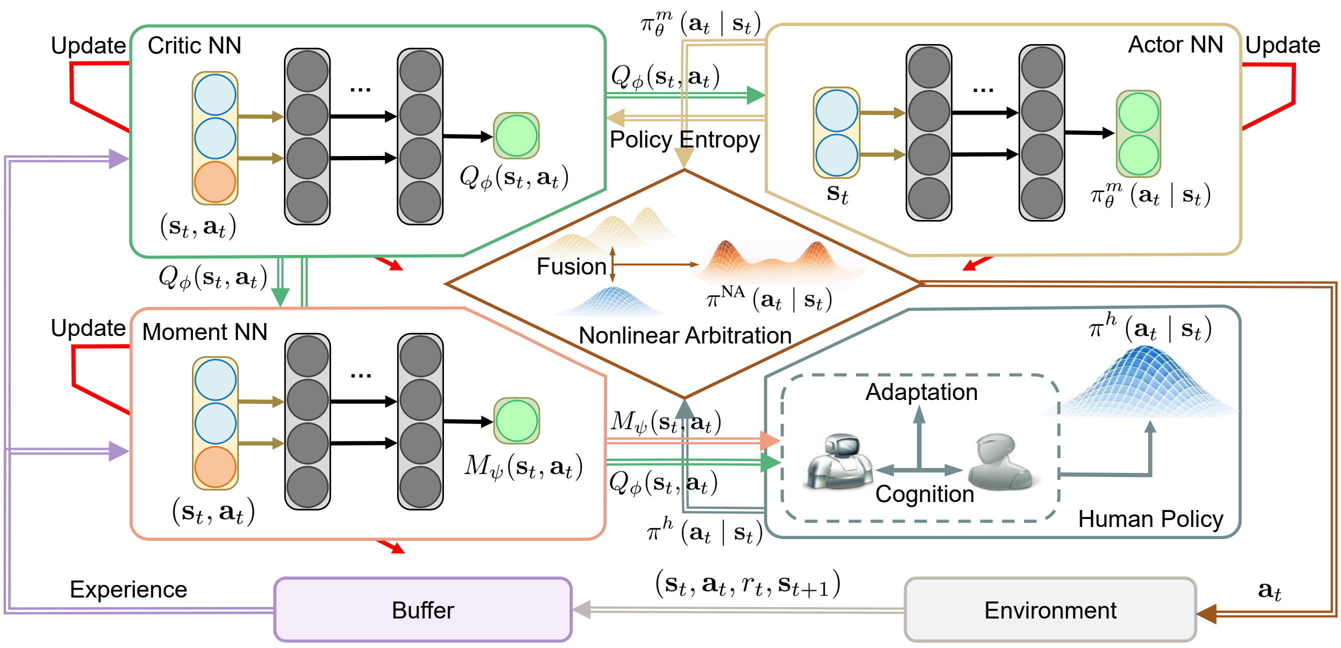
Update
Critic NN
...
$(\mathbf{s}_t, \mathbf{a}_t)$
$Q_\phi(\mathbf{s}_t, \mathbf{a}_t)$
$\pi_\theta^m(\mathbf{a}_t \mid \mathbf{s}_t)$
$Q_\phi(\mathbf{s}_t, \mathbf{a}_t)$
Policy Entropy
Actor NN
Update
$\mathbf{s}_t$
...
$\pi_\theta^m(\mathbf{a}_t \mid \mathbf{s}_t)$
$Q_\phi(\mathbf{s}_t, \mathbf{a}_t)$
Fusion
$\pi^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t)$
Nonlinear Arbitration
Update
Moment NN
...
$M_\psi(\mathbf{s}_t, \mathbf{a}_t)$
$(\mathbf{s}_t, \mathbf{a}_t)$
$M_\psi(\mathbf{s}_t, \mathbf{a}_t)$
$Q_\phi(\mathbf{s}_t, \mathbf{a}_t)$
$\pi^h(\mathbf{a}_t \mid \mathbf{s}_t)$
$\pi^h(\mathbf{a}_t \mid \mathbf{s}_t)$
Adaptation
Cognition
Human Policy
Experience
Buffer
$(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$
Environment
$\mathbf{a}_t$

Fig. 2. Overview of the proposed UNA-SAC framework. The architecture comprises an Actor neural network (NN) for AI policy generation, a Critic NN for value estimation, a Moment NN for modeling return uncertainty, a Human Policy module for generating and adapting human policies, and a Nonlinear Arbitration module that fuses human and AI policies into the final policy $\pi^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t)$. During interaction with the environment, the system produces experience tuples $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$, which are stored in a replay buffer and used to update all network components.

B. Cognition-Inspired Human Policy Adaptation

Cognitive science research suggests that when human operators perceive a high level of uncertainty in machine policies, they tend to reduce the variability of their behavior, thereby exhibiting a more cautious control mode [35]. Empirical studies have also shown [36]–[38] that when machines are perceived as unreliable, operators are more inclined to adopt conservative policies. Based on these findings, we propose the following hypothesis: when the uncertainty of the AI policy increases, the variance of the human policy will decrease accordingly, reflecting a more conservative and restrictive behavioral pattern.

To characterize this mechanism, we estimate the return variances of the AI and human policies separately based on Eq. (16). Let the return variance when the AI selects action $\mathbf{a}_t^m$ under state $\mathbf{s}_t$ be defined as

$$\mathrm{Var}^m(R_t^E \mid \mathbf{s}_t, \mathbf{a}_t^m) = M_\psi(\mathbf{s}_t, \mathbf{a}_t^m) - \left(Q_\phi(\mathbf{s}_t, \mathbf{a}_t^m)\right)^2, \tag{20}$$

while the return variance when the human selects action $\mathbf{a}_t^h$ under the same state

$$\mathrm{Var}^h(R_t^E \mid \mathbf{s}_t, \mathbf{a}_t^h) = M_\psi(\mathbf{s}_t, \mathbf{a}_t^h) - \left(Q_\phi(\mathbf{s}_t, \mathbf{a}_t^h)\right)^2. \tag{21}$$

Based on the ratio of the two variances, we construct the variance of the human policy as

$$\boldsymbol{\sigma}_h^2(\mathbf{s}_t) = \mathrm{Var}^h(R_t^E \mid \mathbf{s}_t, \mathbf{a}_t^h) \cdot \exp\left(-\beta \cdot \frac{\mathrm{Var}^m(R_t^E \mid \mathbf{s}_t, \mathbf{a}_t^m)}{\mathrm{Var}^h(R_t^E \mid \mathbf{s}_t, \mathbf{a}_t^h) + \epsilon}\right), \tag{22}$$

where $\beta > 0$ denotes the human sensitivity to AI uncertainty, and $\epsilon > 0$ is a small constant introduced to avoid division by zero. When AI uncertainty is high, $\boldsymbol{\sigma}_h^2(\mathbf{s}_t)$ decays exponentially, reflecting a compensatory mechanism in which the human policy contracts toward a more conservative behavior.

Finally, the human policy is modeled as a state-dependent multivariate Gaussian distribution, whose mean is the current human action $\mathbf{a}_t^h$ and covariance is $\mathrm{diag}(\boldsymbol{\sigma}_h^2(\mathbf{s}_t))$:

$$\pi^h(\mathbf{a} \mid \mathbf{s}_t) = \mathcal{N}\big(\mathbf{a} \mid \mathbf{a}_t^h, \mathrm{diag}(\boldsymbol{\sigma}_h^2(\mathbf{s}_t))\big), \tag{23}$$

where $\boldsymbol{\sigma}_h^2(\mathbf{s}_t)$ is dynamically adjusted according to AI uncertainty.

In practice, $\mathbf{a}_t^h$ is provided by the learned human policy π_φ^{H} described in Section V-D, and the above variance modulation is used to capture the empirically observed tendency of human operators to behave more conservatively when they perceive higher uncertainty or unreliability in automation.

C. Nonlinear Arbitration

To fuse the human policy $\pi^h(\mathbf{a}_t \mid \mathbf{s}_t)$ with the AI policy $\pi_\theta^m(\mathbf{a}_t \mid \mathbf{s}_t)$, we adopt a nonlinear arbitration form based on the product of distributions:

$$\pi^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t) = \frac{1}{Z(\mathbf{s}_t)} \pi_\theta^m(\mathbf{a}_t \mid \mathbf{s}_t)\, \pi^h(\mathbf{a}_t \mid \mathbf{s}_t), \tag{24}$$

where the normalization factor is given by

$$Z(\mathbf{s}_t) = \int_{\mathcal{A}} \pi^h(\mathbf{a}' \mid \mathbf{s}_t) \pi_\theta^m(\mathbf{a}' \mid \mathbf{s}_t)\, d\mathbf{a}'. \tag{25}$$

Meanwhile, to enhance the expressive capacity of the AI policy under complex state-action mappings, inspired by [39], we represent the actor policy using a Gaussian Mixture Model (GMM):

$$\pi_{\theta}^{m}(\mathbf{a}_t \mid \mathbf{s}_t) = \sum_{k=1}^{K} w_k \cdot \pi_{\theta_k}^{m}(\mathbf{a}_t \mid \mathbf{s}_t), \tag{26}$$

where

$$\pi_{\theta_k}^{m}(\mathbf{a}_t \mid \mathbf{s}_t) = \mathcal{N}\big(\mathbf{a}_t \mid \boldsymbol{\mu}_k(\mathbf{s}_t;\theta_k), \mathrm{diag}\big(\boldsymbol{\sigma}_k^2(\mathbf{s}_t;\theta_k)\big)\big), \tag{27}$$

with K denoting the number of mixture components, w_k representing the weight of the k-th component under state $\mathbf{s}_t$, such that $\sum_{k=1}^{K} w_k = 1$; and $\boldsymbol{\mu}_k(\mathbf{s}_t;\theta_k)$ and $\boldsymbol{\sigma}_k^2(\mathbf{s}_t;\theta_k)$ denoting the mean and variance vector of each component, respectively. In our implementation, the GMM is directly embedded in the SAC actor and trained end-to-end during SAC learning, without any separate pre-training or offline fitting stage, i.e., its mixture parameters are updated via the SAC objective. The k-th component distribution after arbitration is defined as

$$\pi_{\theta_k}^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t) = \frac{1}{Z_k(\mathbf{s}_t)} \pi^{h}(\mathbf{a}_t \mid \mathbf{s}_t)\, \pi_{\theta_k}^{m}(\mathbf{a}_t \mid \mathbf{s}_t), \tag{28}$$

where

$$Z_k(\mathbf{s}_t) = \int_{\mathcal{A}} \pi^{h}(\mathbf{a}' \mid \mathbf{s}_t)\, \pi_{\theta_k}^{m}(\mathbf{a}' \mid \mathbf{s}_t)\, d\mathbf{a}'. \tag{29}$$

Based on the reparameterization trick,

$$\mathbf{a}_t = \boldsymbol{\mu}_k(\mathbf{s}_t;\theta_k) + \boldsymbol{\sigma}_k(\mathbf{s}_t;\theta_k) \odot \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{30}$$

where $\odot$ denotes the element-wise product.

To improve numerical stability and decouple the optimization of the AI policy from the human model, we apply a stop-gradient operation to the human-policy term. In this way, the human policy acts as an exogenous guidance signal during updates, and the objective does not backpropagate into the human model. This ensures that the AI actor is optimized with respect to the intended SAC objective under a fixed human guidance signal at each update step. We denote the stop-gradient operator by $\mathrm{sg}(\cdot)$, then

$$\begin{aligned}\widetilde{\nabla}_{\mathbf{a}_t} \log \pi_{\theta_k}^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t) := {}& \nabla_{\mathbf{a}_t} \log \pi_{\theta_k}^{m}(\mathbf{a}_t \mid \mathbf{s}_t) \\ & + \nabla_{\mathbf{a}_t} \log \pi^{h}\big(\mathrm{sg}(\mathbf{a}_t) \mid \mathbf{s}_t\big).\end{aligned} \tag{31}$$

Accordingly, the modified objective function is given by

$$J(\pi_{\theta_k}^{\mathrm{NA}}) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}, \mathbf{a}_t \sim \pi_{\theta_k}^{\mathrm{NA}}} \left[\alpha \log \pi_{\theta_k}^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t) - Q_{\phi}(\mathbf{s}_t, \mathbf{a}_t)\right], \tag{32}$$

and the stochastic gradient is given by

$$\begin{aligned}\nabla_{\theta_k} J(\pi_{\theta_k}^{\mathrm{NA}}) = {}& \alpha \nabla_{\theta_k} \log \pi_{\theta_k}^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t) \\ & + \Big(\alpha \widetilde{\nabla}_{\mathbf{a}_t} \log \pi_{\theta_k}^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t) \\ & - \nabla_{\mathbf{a}_t} Q_{\phi}(\mathbf{s}_t, \mathbf{a}_t)\Big) \nabla_{\theta_k} \mathbf{a}_t(\theta_k),\end{aligned} \tag{33}$$

where $\alpha = \sum_k \alpha_k w_k$, and the update of the component temperature α_k follows the method in [39]. According to Eq. (11), the TD target is

$$\begin{aligned}y_t = r_t + \gamma\, \mathbb{E}_{\mathbf{a}_{t+1} \sim \pi^{\mathrm{NA}}} \big[& Q_{\phi}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) \\ & - \alpha \log \pi^{\mathrm{NA}}(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}) \big],\end{aligned} \tag{34}$$

and according to Eq. (12), the critic loss function is

$$\mathcal{L}^{Q}(\phi_i) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1}) \sim \mathcal{D}} \left[\left(Q_{\phi_i}(\mathbf{s}_t, \mathbf{a}_t) - y_t\right)^2 \right], \quad i = 1, 2. \tag{35}$$

With these modifications, we achieve the optimization of a human-AI fused control policy under nonlinear arbitration while preserving the original SAC framework.

Remark 1. *This arbitration mechanism captures human dynamic responsiveness to AI policy uncertainty. When the AI policy exhibits high uncertainty in the current state (i.e., when* $\mathrm{Var}^{m}(R_t^{E} \mid \mathbf{s}_t, \mathbf{a}_t^{m})$ *is large), the regulation mechanism causes* $\boldsymbol{\sigma}_h(\mathbf{s}_t)$ *to shrink significantly, making the human policy distribution* $\pi^{h}(\mathbf{a} \mid \mathbf{s}_t)$ *more concentrated. This enhances the dominance of the human policy in nonlinear arbitration. Conversely, when the AI policy is relatively certain,* $\boldsymbol{\sigma}_h(\mathbf{s}_t)$ *increases, leading to higher variance in the human policy distribution. In this case, nonlinear arbitration relies more on the AI policy* π_{θ}^{m}*, resulting in an AI-dominated control mode.*

Remark 2. *The normalized form of the nonlinear arbitration policy in Eq.* (24) *naturally exhibits a centered gradient structure, effectively reducing the variance of gradient estimation without the need for additional variance-reduction baselines. Specifically, according to the chain rule, we have*

$$\nabla_{\theta_k} \log \pi_{\theta_k}^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t) = \nabla_{\theta_k} \log \pi_{\theta_k}^{m}(\mathbf{a}_t \mid \mathbf{s}_t) - \nabla_{\theta_k} \log Z_k(\mathbf{s}_t), \tag{36}$$

where

$$\begin{aligned}\nabla_{\theta_k} \log Z_k(\mathbf{s}_t) = {}& \frac{1}{Z_k(\mathbf{s}_t)} \int_{\mathcal{A}} \pi^{h}(\mathbf{a} \mid \mathbf{s}_t)\, \pi_{\theta_k}^{m}(\mathbf{a} \mid \mathbf{s}_t) \\ & \cdot \nabla_{\theta_k} \log \pi_{\theta_k}^{m}(\mathbf{a} \mid \mathbf{s}_t)\, d\mathbf{a} \\ = {}& \mathbb{E}_{\mathbf{a}_t \sim \pi_{\theta_k}^{\mathrm{NA}}} \left[\nabla_{\theta_k} \log \pi_{\theta_k}^{m}(\mathbf{a}_t \mid \mathbf{s}_t)\right].\end{aligned} \tag{37}$$

Thus, Eq. (36) *can be further written as*

$$\begin{aligned}\nabla_{\theta_k} \log \pi_{\theta_k}^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t) = {}& \nabla_{\theta_k} \log \pi_{\theta_k}^{m}(\mathbf{a}_t \mid \mathbf{s}_t) \\ & - \mathbb{E}_{\mathbf{a} \sim \pi_{\theta_k}^{\mathrm{NA}}} \left[\nabla_{\theta_k} \log \pi_{\theta_k}^{m}(\mathbf{a} \mid \mathbf{s}_t)\right],\end{aligned} \tag{38}$$

explicitly revealing the mean-centering property of the gradient.

Remark 3. *When the AI policy is a unimodal Gaussian (*$K = 1$*), the fusion result further degenerates into linear arbitration. Specifically, for the k-th Gaussian component of the AI policy, the new Gaussian distribution parameters obtained after multiplication with the human policy are given by*

$$\boldsymbol{\Sigma}_k^{\mathrm{NA}}(\mathbf{s}_t) = \Big[\mathrm{diag}(\boldsymbol{\sigma}_k^2(\mathbf{s}_t;\theta_k))^{-1} + \mathrm{diag}(\boldsymbol{\sigma}_h^2(\mathbf{s}_t))^{-1}\Big]^{-1}, \tag{39}$$

$$\begin{aligned}\boldsymbol{\mu}_k^{\mathrm{NA}}(\mathbf{s}_t) = \boldsymbol{\Sigma}_k^{\mathrm{NA}}(\mathbf{s}_t) \Big[& \mathrm{diag}(\boldsymbol{\sigma}_k^2(\mathbf{s}_t;\theta_k))^{-1} \boldsymbol{\mu}_k(\mathbf{s}_t;\theta_k) \\ & + \mathrm{diag}(\boldsymbol{\sigma}_h^2(\mathbf{s}_t))^{-1} \mathbf{a}_t^{h}\Big].\end{aligned} \tag{40}$$

Accordingly, the nonlinear arbitration policy can be written as

$$\pi^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t) = \sum_{k=1}^{K} \tilde{w}_k \cdot \mathcal{N}\big(\mathbf{a}_t \mid \boldsymbol{\mu}_k^{\mathrm{NA}}(\mathbf{s}_t), \boldsymbol{\Sigma}_k^{\mathrm{NA}}(\mathbf{s}_t)\big), \tag{41}$$

Algorithm 1 UNA-SAC Algorithm

```
Initialize critic parameters $\phi$;  target critic $\bar{\phi} \leftarrow \phi$
Initialize actor parameters $\{\theta_k, \alpha_k\}_{k=1}^{K}$
Initialize mixture-weight parameters $[w_1, w_2, \ldots, w_K]$
Initialize moment parameters $\psi$
Initialize replay buffer $\mathcal{D} \leftarrow \emptyset$
Load pre-trained human model parameters $\varphi$
for each iteration do
    for each environment step do
        Observe state $\mathbf{s}_t$
        Sample mixture component $i \sim [w_1, w_2, \ldots, w_K]$
        Sample AI action $\mathbf{a}_t^m \sim \pi_{\theta_i}^m(\mathbf{a}_t \mid \mathbf{s}_t)$
        Sample human action $\mathbf{a}_t^h \sim \pi_{\varphi}^{\mathrm{H}}(\mathbf{a}_t \mid \mathbf{s}_t)$
        Estimate uncertainties via Eq. (20) and Eq. (21)
        Build cognition-adjusted human policy via Eq. (23)
        Form nonlinear arbitration policy $\pi_{\theta_i}^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t)$ using Eq. (28)
        Sample final action $\mathbf{a}_t \sim \pi_{\theta_i}^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t)$
        Execute $\mathbf{a}_t$, observe $\mathbf{r}_t$, $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$
        Store $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$ in $\mathcal{D}$
    end for
    for each gradient step do
        Update critic parameters $\phi$ using Eq. (35)
        $\bar{\phi} \leftarrow \tau\phi + (1-\tau)\bar{\phi}$
        for $i = 1$ to $K$ do
            Update actor component $\theta_i$ using Eq. (33)
        end for
        Update moment parameters $\psi$ using Eq. (19)
    end for
    if a new human demonstration arrives then
        Store the demonstration into $\mathcal{D}_{\mathcal{H}}$
        Update human model $\varphi$ by minimizing Eq. (47) with samples from $\mathcal{D}_{\mathcal{H}}$
    end if
end for
```

where $\tilde{w}_k$ denotes the normalized weight of the fused component. When $K = 1$, we have

$$\boldsymbol{\Sigma}^{\mathrm{NA}}(\mathbf{s}_t) = \left[\mathrm{diag}(\boldsymbol{\sigma}_1^2(\mathbf{s}_t;\theta_1))^{-1} + \mathrm{diag}(\boldsymbol{\sigma}_h^2(\mathbf{s}_t))^{-1}\right]^{-1}, \tag{42}$$

$$\boldsymbol{\mu}^{\mathrm{NA}}(\mathbf{s}_t) = \boldsymbol{\Sigma}^{\mathrm{NA}}(\mathbf{s}_t)\Big[\mathrm{diag}(\boldsymbol{\sigma}_1^2(\mathbf{s}_t;\theta_1))^{-1}\boldsymbol{\mu}_1(\mathbf{s}_t;\theta_1) + \mathrm{diag}(\boldsymbol{\sigma}_h^2(\mathbf{s}_t))^{-1}\boldsymbol{\mu}^h(\mathbf{s}_t)\Big]. \tag{43}$$

If the fused mean $\boldsymbol{\mu}^{\mathrm{NA}}(\mathbf{s}_t)$ is directly taken as the output action, the result is equivalent to linear arbitration:

$$\pi^{\mathrm{LA}}(\mathbf{a}_t \mid \mathbf{s}_t) = \lambda_t \cdot \pi^h(\mathbf{a}_t \mid \mathbf{s}_t) + (1-\lambda_t)\cdot \pi_\theta^m(\mathbf{a}_t \mid \mathbf{s}_t), \tag{44}$$

where $\lambda_t \in [0,1]$ denotes the weight of human control authority.

D. Human Modeling

To reduce the burden on human participants during training, we construct a human model to learn and simulate real human behavior in collaborative tasks, thereby substituting for direct human control in certain scenarios. In our implementation, this human policy model is constructed in two stages. First, we collect offline demonstrations from five human drivers in the driving-assistance scenario and aggregate their trajectories to build the demonstration dataset $\mathcal{D}_{\mathcal{H}}$ used for imitation learning. Then, one of the drivers provides online corrections during interactive training to further adjust the human policy model.

We adopt an online imitation learning update mechanism [40] to model human outputs with a policy network $\pi_\varphi^{\mathrm{H}}(\mathbf{a} \mid \mathbf{s})$ by minimizing the discrepancy between $\pi_\varphi^{\mathrm{H}}(\mathbf{a} \mid \mathbf{s})$ and the empirical distribution of human outputs $\pi^h(\mathbf{a} \mid \mathbf{s})$:

$$\varphi^* = \arg\min_{\varphi} \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}_{\mathcal{H}}}\left[d\left(\pi^h(\cdot \mid \mathbf{s}_t),\, \pi_\varphi^{\mathrm{H}}(\cdot \mid \mathbf{s}_t)\right)\right], \tag{45}$$

where $d(\cdot,\cdot)$ denotes the policy discrepancy measure, which in this work is instantiated as the squared ℓ_2 distance between the human action and the model output, as in Eq. (47), and $\mathcal{D}_{\mathcal{H}}$ represents the dataset of human demonstrations. This online imitation learning scheme is particularly suitable for our shared control setting. First, by continuously updating the human policy model on the states actually visited under UNA-SAC, the online imitation learning scheme adapts the model to the on-policy state distribution, rather than being restricted to the offline demonstration distribution. This is particularly important in shared control, where the state distribution changes as the arbitration policy evolves during training. Second, it reduces the burden on human participants: instead of providing full-time manual control throughout training, the driver only needs to provide corrective actions when the current model deviates from the desired behavior, and these corrections are incrementally distilled into π_φ^H. Compared with purely offline imitation learning using a fixed demonstration dataset, the online update mitigates distribution shift. Compared with requiring full-time human control, it substantially reduces human workload.

During the initialization phase of the human policy model, we set its structure to be identical to that of the untrained actor network and initialize its parameters using a reference policy π_{ref}:

$$\pi_{\varphi_0}^{\mathrm{H}}(\mathbf{a}_t \mid \mathbf{s}_t) \leftarrow \pi_{\text{ref}}(\mathbf{a}_t \mid \mathbf{s}_t). \tag{46}$$

After each interaction, the parameters are updated based on the newly collected human trajectory data $(\mathbf{s}_t, \mathbf{a}_t^h)$, with the loss function defined as:

$$\mathcal{L}^{\mathrm{H}}(\varphi) = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t^h)\sim\mathcal{D}_{\mathcal{H}}}\left[\left\|\mathbf{a}_t^h - \pi_\varphi^{\mathrm{H}}(\mathbf{s}_t)\right\|_2^2\right], \tag{47}$$

where $\|\cdot\|_2$ denotes the ℓ_2 norm of a vector. As the iterations proceed, the model π_φ^{H} gradually approaches the real human policy and provides auxiliary action decisions in subsequent tasks, thereby reducing the need for frequent human involvement during training.

In the subsequent experiments, this learned policy π_φ^{H} acts as a surrogate for real human drivers: its outputs are used as the human commands in the shared control loop, while its mean behaviour is anchored to human demonstrations and its variability is further shaped by the uncertainty-aware adaptation mechanism described in Section V-B.

E. UNA-SAC Algorithm

To provide a more intuitive illustration of the proposed UNA-SAC implementation, this section presents the algorithmic pseudocode in Algorithm 1. During the initialization phase, UNA-SAC incorporates a pre-trained human model. In the interaction phase, it estimates the uncertainty of both AI and human actions, constructs a cognition-driven human policy, and generates the final action through the nonlinear arbitration mechanism. In the update phase, the algorithm not only updates the actor-critic networks and the moment network but also continuously refines the human model using newly collected human demonstrations. These designs ensure that the algorithm can better integrate human knowledge and enhance policy robustness under uncertain environments.

VI. Comparison of Policy Gradients

In policy optimization for shared control, the structure of the arbitration mechanism directly affects the direction and stability of gradient updates. If the arbitration amplifies biases from external uncertainty estimates, these errors will accumulate over multiple iterations, causing the optimization trajectory to deviate from the optimal solution and even leading to convergence failure. Traditional linear arbitration policies in Eq. (44) rely on explicit uncertainty weighting; when estimates are inaccurate, the bias is directly propagated into gradient computation, thereby magnifying its interference with the optimization process. In contrast, the nonlinear arbitration policy in Eq. (24) fuses policies through the product of distributions, without relying on external weight estimation, and structurally mitigates the influence of uncertainty bias on the update direction.

To quantitatively characterize this difference, we derive an upper bound on the gradient discrepancy between the nonlinear arbitration policy and the AI policy π_θ^m based on the reparameterized form of the maximum entropy policy gradient, leading to the following theorem.

Theorem 1. *Let the state space $\mathcal{S}$ and the action space $\mathcal{A}$ be compact sets. Assume there exist constants $C_1 > 0$, $C_2 > 0$, and $C_3 > 0$ such that*

$$\|\nabla_\theta f_\theta\|_\infty := \sup_{\mathbf{s}_t \in \mathcal{S},\, \mathbf{a}_t \in \mathcal{A}} \|\nabla_\theta f_\theta(\mathbf{a}_t, \mathbf{s}_t)\| \leq C_1, \tag{48}$$

$$\|\nabla_{\mathbf{a}_t} \log \pi_\theta^m\|_\infty \leq C_2, \quad \|\nabla_{\mathbf{a}_t} Q_\phi\|_\infty \leq C_3. \tag{49}$$

If the ℓ_1 norm between the nonlinear arbitration policy $\pi^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t)$ and the AI policy $\pi_\theta^m(\mathbf{a}_t \mid \mathbf{s}_t)$ satisfies

$$\|\pi^{\mathrm{NA}} - \pi_\theta^m\|_1 := \sup_{\mathbf{s} \in \mathcal{S}} \int_{\mathcal{A}} \left|\pi^{\mathrm{NA}}(\mathbf{a}_t \mid \mathbf{s}_t) - \pi_\theta^m(\mathbf{a}_t \mid \mathbf{s}_t)\right| d\mathbf{a}_t \leq \epsilon, \tag{50}$$

and there exists a function $\kappa(\epsilon)$ such that $\lim_{\epsilon \to 0} \kappa(\epsilon) = 0$ and $\|\nabla_{\mathbf{a}_t} \log \pi^h\|_\infty \leq \kappa(\epsilon)$, then there exists

$$\delta(\epsilon) = \alpha C_1 \kappa(\epsilon) + C_1(\alpha C_2 + C_3)\,\epsilon, \tag{51}$$

such that

$$|\nabla_\theta J(\pi^{\mathrm{NA}}) - \nabla_\theta J(\pi_\theta^m)| \leq \delta(\epsilon), \tag{52}$$

and moreover $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.

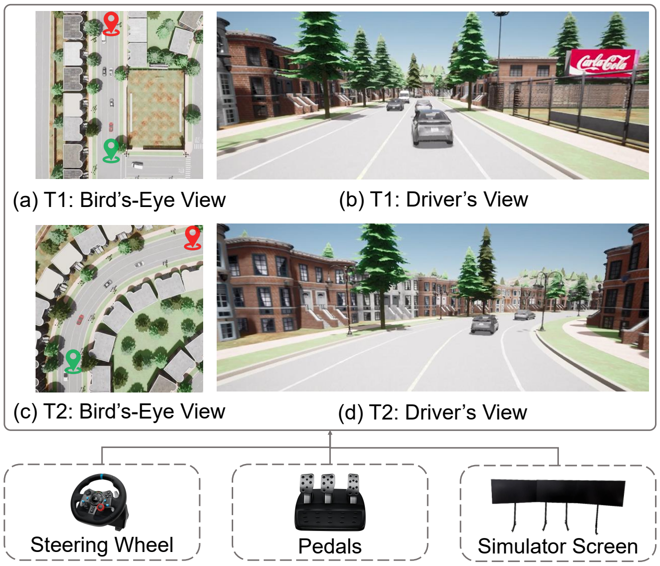
(a) T1: Bird's-Eye View
(b) T1: Driver's View
(c) T2: Bird's-Eye View
(d) T2: Driver's View

Fig. 3. Visualization of driving tasks. Subplots (a) and (c) show the bird's-eye view of Task 1 (T1) and Task 2 (T2), while (b) and (d) present the corresponding driver's view.

Proof. See Appendix A.

Remark 4. *With the stop-gradient operation, we have $\kappa(\epsilon) \equiv 0$, in which case the first term in the theorem vanishes.*

This result indicates that when the nonlinear arbitration policy in Eq. (24) is close to the AI policy distribution, its gradient direction is almost aligned with that of the AI policy, thereby avoiding the introduction of additional optimization bias. In contrast, the linear arbitration policy in Eq. (44) relies on the external uncertainty estimate through λ_t, and its weighted-sum structure imposes a lower bound on the ℓ_1 norm between π^{LA} and π_θ^m. As a result, the closeness condition in Theorem 1 cannot be satisfied, and the convergence of the gradient discrepancy cannot be guaranteed.

VII. Experimental Verification

In this section, we construct a driving assistance scenario based on the CARLA simulation platform to evaluate the performance of the proposed UNA-SAC method in human-AI co-driving within traffic environments. The experimental environment is deployed on Ubuntu 20.04 with CARLA 0.9.14 using the Town05 map, and the hardware configuration includes an NVIDIA RTX 4090 GPU. To ensure fairness and reproducibility, all models are trained and evaluated under identical system configurations and parameter settings. Specifically, both the actor and critic networks in UNA-SAC are implemented as two-layer fully connected neural networks, each with 256 hidden units and ReLU activations. The discount factor is set to $\gamma = 0.99$, the soft update coefficient to $\tau = 0.005$, and the replay buffer capacity to 10^6. The Adam optimizer is employed with a learning rate of 3×10^{-4}. Training was repeated independently with five different random seeds to ensure statistical robustness. During evaluation, each trained model was tested for 50 episodes, and the final performance was reported as the mean across these runs.

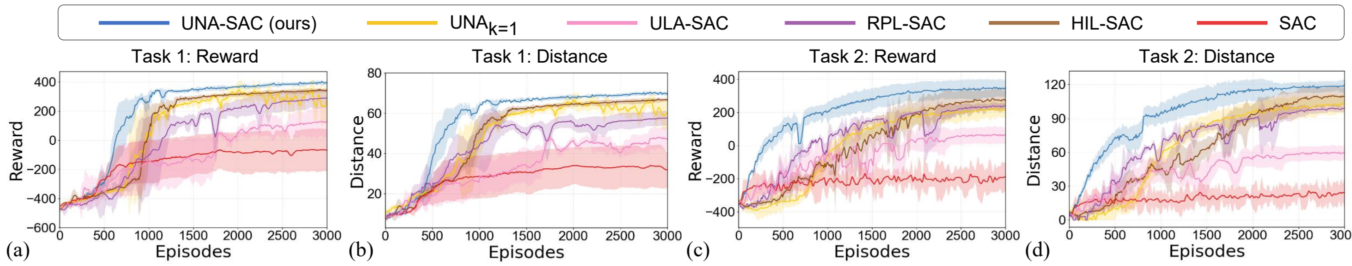

Fig. 4. Comparison of training performance across different arbitration policies in Task 1 and Task 2. Subplots (a) and (b) show the Reward and Distance for Task 1, while subplots (c) and (d) show the Reward and Distance for Task 2.

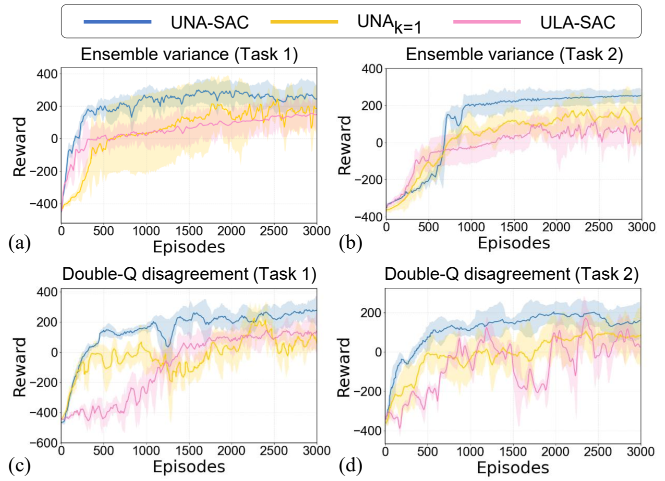

Fig. 5. Comparison of training performance under different uncertainty estimators, with subplots (a) and (c) showing Task 1 and subplots (b) and (d) showing Task 2. Subplots (a) and (b) use ensemble variance [41], while subplots (c) and (d) use double-Q disagreement [42].

A. Experimental Settings

In the driving assistance scenario, the vehicle is jointly controlled by a human driver and the AI decision module, with the human retaining primary authority. The control actions are represented as two-dimensional continuous vectors corresponding to lateral and longitudinal control. The reward function is primarily designed to promote lane keeping and speed stability, while also optimizing directional consistency, trajectory foresight, and steering smoothness, with an additional acceleration reward provided during the starting phase. At the same time, penalties are imposed for abrupt steering, counter-steering, severe yaw, lane departure, and collisions to ensure driving safety and stability.

To further evaluate the adaptability of the proposed method under different driving situations, we select two representative driving tasks. Task 1 is straight driving, where the vehicle, under the primary control of the human driver, moves from the green starting point to the red endpoint along a straight road segment. Task 2 is curved road driving, where the vehicle also travels from the green starting point to the red endpoint but passes through a curved road segment. In both scenarios, the system continuously perceives human operations and assists the driver in maintaining stable driving by adjusting steering and longitudinal control. In these experiments, we use the human policy model trained in Section V-D to simulate human inputs. The experimental platform consists of a steering wheel, pedals, and a simulation screen, as shown in Fig. 3; subplots (a)-(d) illustrate the bird's-eye views and driver's views of the two representative driving scenarios.

B. Baseline Algorithms

To comprehensively evaluate the effectiveness of the proposed nonlinear arbitration policy, we select five representative baselines, which cover different paradigms including pure autonomous control, linear arbitration mechanisms, distribution-based arbitration, and residual policy learning.

1) SAC [18]: A continuous control algorithm based on maximum entropy policy optimization, which is a widely used and efficient reinforcement learning method. This baseline does not incorporate any human information and serves as a reference for pure autonomous control, providing a benchmark for evaluating the optimal performance without human intervention.

2) ULA-SAC [13]–[15], [25], [26]: A shared control method based on linear arbitration, which fuses human and AI control actions through weighted averaging to achieve collaborative decision-making. We implement this mechanism within the SAC framework and incorporate the same moment network and human cognitive adaptation policy as in our proposed method to ensure consistency in comparison.

3) HIL-SAC [27]: Human-AI collaborative decision-making guided by the value function, which dynamically allocates control authority during execution by selecting actions that are both high-valued and close to human inputs. The original method is based on DQN, and in this study we re-implement it within the SAC framework for a unified comparison.

4) RPL-SAC [29]: The Residual Policy Learning (RPL) method learns a minimally invasive human-AI collaboration policy without relying on environment models or user goals. The original method is implemented with PPO, while in this study we adapt it to the SAC framework to ensure consistency.

5) UNA$_{K=1}$: A degenerate version of the proposed method, where the fused policy in each dimension is modeled using a single Gaussian distribution (i.e., the number of Gaussian mixture components $K = 1$). According to Remark 3, this is equivalent to the linear arbitration policy and is used to evaluate the performance gain of the nonlinear policy under the same conditions.

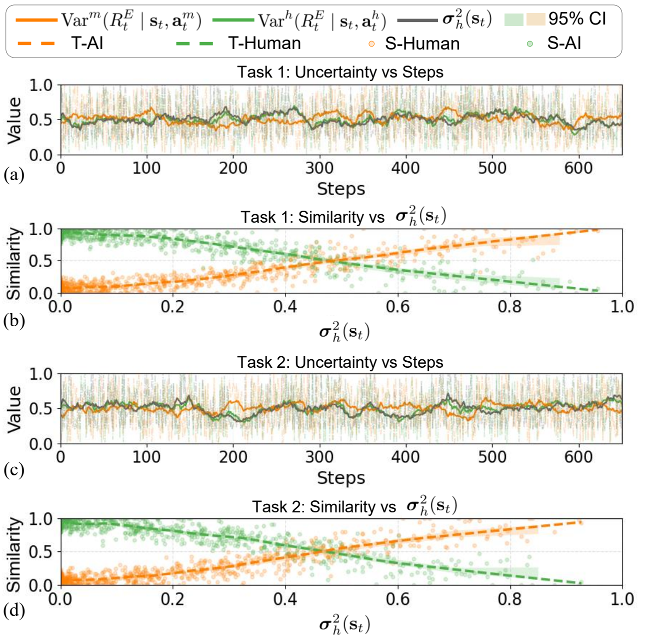

Fig. 6. Uncertainty estimation and similarity analysis, with Task 1 in subplots (a) and (b) and Task 2 in subplots (c) and (d). Subplots (a) and (c) show the uncertainty variation between human and AI policies during execution. Subplots (b) and (d) show the relationship between human-AI similarity and $\boldsymbol{\sigma}_h^2(\mathbf{s}_t)$. S-Human and S-AI indicate the similarity with human and AI policies, respectively. T-Human and T-AI are LOWESS trend lines with shaded areas representing 95% confidence intervals.

C. Training Performance

To comprehensively evaluate the training performance of the proposed nonlinear arbitration method, this section analyzes two perspectives: (i) comparing the performance of different arbitration policies under unified experimental conditions, and (ii) examining their adaptability under different uncertainty estimation methods.

1) Performance under Different Arbitration Policies: We compare the training performance of different arbitration policies in Task 1 and Task 2, using the cumulative reward per episode (Reward) and the driving distance before task termination (Distance) as evaluation metrics, where higher values indicate better performance.

Fig. 4 presents the training curves, where the solid lines denote the mean performance over five runs and the shaded areas represent the standard deviation. In Task 1 (Figs. 4(a) and 4(b)), UNA-SAC achieves the best performance in terms of convergence speed, final cumulative reward, and driving distance, with relatively small variance. The degenerate linear version $\mathrm{UNA}_{\mathrm{K=1}}$ converges faster in the early stage; however, once entering the stable phase, its curves exhibit larger fluctuations, indicating less stable convergence. ULA-SAC shows a lower convergence rate during training, with limited levels of cumulative reward and driving distance. RPL-SAC improves at a moderate pace and converges relatively early, but its overall performance remains low and fails to reach a higher performance ceiling. HIL-SAC converges quickly in the early stage, with final performance lying between $\mathrm{UNA}_{\mathrm{K=1}}$ and RPL-SAC. Standard SAC, without human input, struggles to handle complex tasks, and its overall performance is significantly inferior to that of human-AI fusion policies. In Task 2 (Figs. 4(c) and 4(d)), UNA-SAC again achieves the highest cumulative reward and driving distance, verifying its adaptability and stability across different task scenarios. Overall, the proposed UNA-SAC method integrates human and AI policies more effectively, achieving significantly better learning efficiency and task performance than linear arbitration and other baselines.

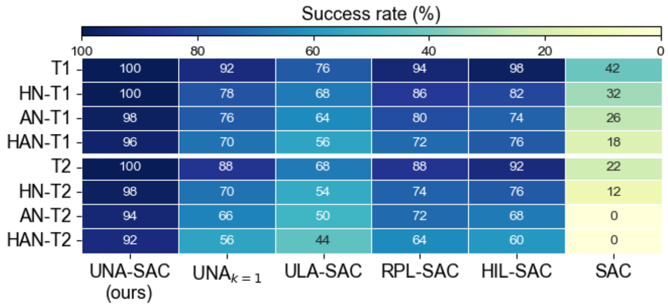

Fig. 7. Heatmaps of success rates (%) across methods. HN: human noise (added to the human policy output); AN: AI policy noise (added to the AI policy output); HAN: human + AI policy noise (added to both stages).

2) Performance under Different Uncertainty Estimation Methods: We further investigate the adaptability of UNA-SAC under different uncertainty estimation methods. Two commonly used methods are adopted: ensemble variance [41], which quantifies uncertainty through the variance of value predictions from critic networks, and double-Q disagreement [42], which characterizes uncertainty by the magnitude of the difference between value estimates from two critic networks. In the experiments, the algorithmic structure and hyperparameters are held fixed, with only the uncertainty estimation method being replaced.

Fig. 5 shows the training results. For the ensemble variance metric (Figs. 5(a) and 5(b)), UNA-SAC achieves the fastest convergence and the highest final cumulative reward in both tasks; $\mathrm{UNA}_{\mathrm{K=1}}$ converges slightly slower with marginally lower performance; ULA-SAC suffers from insufficient convergence accuracy. Under double-Q disagreement (Figs. 5(c) and 5(d)), UNA-SAC still maintains superiority in convergence, performance, and stability; $\mathrm{UNA}_{\mathrm{K=1}}$ performs comparably in the early stage but lags behind later; ULA-SAC remains consistently lower with significant fluctuations. Overall, the proposed UNA-SAC method retains its advantage across different uncertainty estimation metrics, validating its robustness.

D. Testing Performance

To systematically evaluate the arbitration policy after training, this section presents analysis from two perspectives: (i) mechanism verification and (ii) performance evaluation.

1) Mechanism Validation: In the mechanism verification part, we focus on examining the effectiveness of the cognition-driven human policy adaptation mechanism and the nonlinear arbitration mechanism, in order to verify whether they can dynamically respond to changes in uncertainty during operation as theoretically designed.

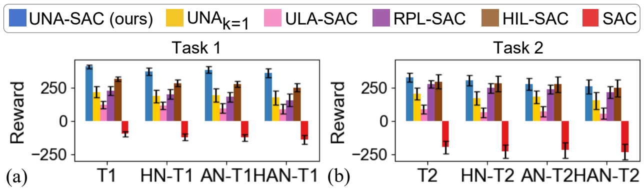

Fig. 8. Barplots of average rewards of different methods in (a) Task 1 and (b) Task 2 under various noise conditions. The error bars represent the standard deviation.

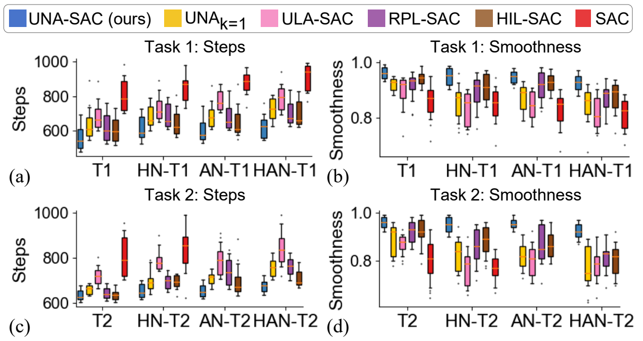

Fig. 9. Boxplots of steps and smoothness under successful episodes in Task 1 and Task 2 with various noise conditions. Subplots (a) and (b) show the steps and smoothness for Task 1, and subplots (c) and (d) show the steps and smoothness for Task 2.

We first verify the implementation of the cognition-driven human policy adaptation mechanism proposed in Section V-B. To this end, we examine the trends of the AI action return variance $\mathrm{Var}^m(R_t^E \mid s_t, a_t^m)$, the human action return variance $\mathrm{Var}^h(R_t^E \mid s_t, a_t^h)$, and the human policy variance $\sigma_h^2(\mathbf{s}_t)$ during execution, in order to characterize the impact of AI action uncertainty on the variance of the human policy across different states. Subplots (a) and (c) of Fig. 6 show that when the AI policy uncertainty is high, the human policy reduces its variance to maintain stability of control authority, thereby confirming that UNA-SAC successfully realizes this cognitive hypothesis.

We then investigate the response characteristics of the nonlinear arbitration mechanism proposed in Section V-C with respect to variations in human policy variance. Specifically, the inverse-normalized ℓ_2 norm is employed to measure the differences between the arbitration action of UNA-SAC and those of the human and AI policies, thereby computing the similarity between the arbitration policy and the human policy (S-Human), as well as with the AI policy (S-AI). Combined with LOWESS smoothed trend lines (T-Human and T-AI) and their 95% confidence intervals, we analyze how the similarity evolves with respect to $\sigma_h^2(\mathbf{s}_t)$. Subplots (b) and (d) of Fig. 6 demonstrate that when the human policy variance is low, the arbitration policy exhibits significantly higher similarity to the human policy, whereas when the variance is high, the arbitration policy gradually shifts toward the AI policy. This result indicates that nonlinear arbitration can dynamically adjust according to AI uncertainty, in line with theoretical expectations, thereby further validating the effectiveness of the proposed model.

2) Performance Evaluation: In the performance evaluation part, we systematically assess the proposed UNA-SAC method under different noise disturbance conditions from three perspectives: task success rate, average cumulative reward, and execution characteristics, thereby validating its capability to capture and adapt to uncertainty information.

We inject zero-mean Gaussian noise with a standard deviation of 5% into the control loop, with three types of disturbance settings: HN (Human Noise), where noise is added to the actions output by the human policy; AN (AI policy Noise), where noise is added to the actions sampled from the AI policy; and HAN (Human + AI policy Noise), where noise is applied to both of the above stages simultaneously. In each scenario, we perform 50 test runs using the same random seeds, and a task is considered successful if the ego vehicle is within 5 meters of the target point at the end of the task.

Fig. 7 presents the success rate comparison of different methods in Task 1 and Task 2. The results show that UNA-SAC achieves the highest success rates across all test scenarios. In particular, it maintains a significant advantage under human noise, AI policy noise, and combined disturbances, demonstrating strong robustness and adaptability. In contrast, its degenerate version UNA$_{K=1}$ performs comparably to UNA-SAC in some scenarios but exhibits a clear decline in success rate under high-noise disturbances, indicating that the nonlinear fusion structure provides greater advantages in complex interference conditions. ULA-SAC shows a general drop in success rate under noise disturbances, reflecting its limited adaptability to uncertainty. HIL-SAC and RPL-SAC maintain a certain level of performance under low-noise conditions but degrade significantly in HAN scenarios. Standard SAC consistently yields the lowest success rates across all complex settings, dropping below 30% in AN-T1 and HAN-T1, and even failing completely in AN-T2 and HAN-T2, which highlights the difficulty of handling complex driving tasks without the assistance of human policies.

After verifying the advantage in task success rates, we further analyze the performance of each method under different noise conditions from the perspective of cumulative reward, in order to evaluate their stability and overall control quality.

Fig. 8 shows the average cumulative rewards of different methods in Task 1 and Task 2 under various noise conditions. It can be observed that UNA-SAC achieves the highest average cumulative rewards across all noise settings in both tasks, with relatively small standard deviations, demonstrating strong stability and robustness. In contrast, the methods with linear arbitration structures, UNA$_{K=1}$ and ULA-SAC, exhibit slightly lower overall cumulative reward levels, indicating the advantage of nonlinear arbitration in suppressing noise disturbances. Other baseline methods also show significant degradation under noisy conditions, further confirming that UNA-SAC is more effective in mitigating disturbances and maintaining reliable control.

However, high success rates and high cumulative rewards alone cannot fully capture the characteristics of the driving process. Therefore, we further conduct a comparative analysis from two additional perspectives: task execution efficiency and action smoothness.

Fig. 9(a) and (c) show the survival steps in successful episodes for each method. The results indicate that UNA-SAC consistently requires significantly fewer steps than other methods under all noise conditions, achieving the shortest task completion time while maintaining a low variance even in high-noise scenarios. In contrast, although the other baseline methods can complete the tasks under noise-free conditions, they generally require more steps; under noisy disturbances, the number of steps further increases, with particularly large fluctuations under HAN, suggesting that these methods struggle to maintain efficient control in highly disturbed environments.

Fig. 9(b) and (d) present the smoothness metric, which is computed based on the second-order differences of actions. UNA-SAC achieves smoothness values close to 1.0 across both tasks and all noise conditions, with highly concentrated distributions and minimal variance, demonstrating superior stability and driving comfort. In contrast, while other baseline methods maintain a certain degree of smoothness under noise-free conditions, their overall levels remain lower than those of UNA-SAC; under noisy scenarios, their smoothness drops significantly with larger fluctuations, making it difficult to ensure stable trajectory control in highly disturbed environments.

VIII. Conclusion

This paper addresses the impact of AI policy uncertainty on shared control arbitration and proposes a nonlinear arbitration method, UNA-SAC. The method introduces a moment network to model AI uncertainty and employs a cognition-driven human policy adaptation mechanism to dynamically adjust the human policy distribution. At the distributional level, it constructs a nonlinear product fusion of human and AI policies, fundamentally mitigating the cumulative distortion caused by uncertainty estimation bias in linear arbitration. Theoretical analysis demonstrates that the proposed method ensures stability in gradient updates. Experimental results further validate the effectiveness of UNA-SAC in driving assistance scenarios: compared with baseline methods, it achieves significant improvements in convergence efficiency, task reliability, robustness, and operational performance.

Beyond the specific driving-assistance setting, the proposed UNA-SAC framework provides a general approach for integrating uncertainty-aware nonlinear arbitration into AI-powered human—machine systems. Future research may extend this framework to other shared control domains such as robotic manipulation and assistive robotics, incorporate richer models of human behaviour and trust, and combine UNA-SAC with alternative reinforcement learning backbones to further enhance safety and robustness in safety-critical applications. Moreover, we will extend the interactive correction stage to incorporate corrections from multiple drivers, thereby improving the generalizability and robustness of the learned human policy adaptation.

Appendix A
Proof of Theorem 1

From Eq. (7), the maximum entropy policy gradient can be decomposed into an explicit term and a pathwise term. In expectation, the explicit term satisfies

$$\mathbb{E}_{\mathbf{s}_t,\,\boldsymbol{\epsilon}_t}\big[\nabla_\theta(\alpha\log\pi_\theta(\mathbf{a}_t\mid\mathbf{s}_t))\big]=0, \tag{A.1}$$

and therefore only the pathwise term needs to be retained:

$$\begin{aligned}\nabla_\theta J(\pi)=\mathbb{E}_{\mathbf{s}_t,\,\boldsymbol{\epsilon}_t}\Big[\quad&\big(\alpha\nabla_{\mathbf{a}_t}\log\pi(\mathbf{a}_t\mid\mathbf{s}_t)\\&-\nabla_{\mathbf{a}_t}Q_\phi(\mathbf{s}_t,\mathbf{a}_t))\Big|_{\mathbf{a}_t=f_\theta(\boldsymbol{\epsilon}_t;\mathbf{s}_t)}\\&\times\nabla_\theta f_\theta(\boldsymbol{\epsilon}_t;\mathbf{s}_t)\Big].\end{aligned} \tag{A.2}$$

Define

$$G_\pi(\mathbf{s}_t,\mathbf{a}_t):=\alpha\,\nabla_{\mathbf{a}_t}\log\pi(\mathbf{a}_t\mid\mathbf{s}_t)-\nabla_{\mathbf{a}}Q_\phi(\mathbf{s}_t,\mathbf{a}_t), \tag{A.3}$$

$$J_f(\mathbf{s}_t,\boldsymbol{\epsilon}_t):=\nabla_\theta f_\theta(\boldsymbol{\epsilon}_t;\mathbf{s}_t). \tag{A.4}$$

From Eq. (A.2), the gradient difference is

$$\begin{aligned}\Delta&:=\nabla_\theta J(\pi^{\mathrm{NA}})-\nabla_\theta J(\pi_\theta^m)\\&=\mathbb{E}_{\mathbf{s}_t,\,\boldsymbol{\epsilon}_t}\big[G_{\pi^{\mathrm{NA}}}(\mathbf{s}_t,\mathbf{a}_{\mathrm{NA}})-G_{\pi_\theta^m}(\mathbf{s}_t,\mathbf{a}_m)\big]\,J_f(\mathbf{s}_t,\boldsymbol{\epsilon}_t)\end{aligned} \tag{A.5}$$

where $\mathbf{a}_{\mathrm{NA}}$ and $\mathbf{a}_m$ are sampled from $\pi^{\mathrm{NA}}(\mathbf{a}_t\mid\mathbf{s}_t)$ and $\pi_\theta^m(\mathbf{a}_t\mid\mathbf{s}_t)$, respectively. Eq. (A.5) can be decomposed into two terms:

$$\Delta_1=\mathbb{E}_{\mathbf{s}_t,\boldsymbol{\epsilon}_t}\Big[\big(G_{\pi^{\mathrm{NA}}}-G_{\pi_\theta^m}\big)(\mathbf{s}_t,\mathbf{a}_{\mathrm{NA}})\,J_f(\mathbf{s}_t,\boldsymbol{\epsilon}_t)\Big], \tag{A.6}$$

$$\Delta_2=\mathbb{E}_{\mathbf{s}_t,\boldsymbol{\epsilon}_t}\Big[\big(G_{\pi_\theta^m}(\mathbf{s}_t,\mathbf{a}_{\mathrm{NA}})-G_{\pi_\theta^m}(\mathbf{s}_t,\mathbf{a}_m)\big)\,J_f(\mathbf{s}_t,\boldsymbol{\epsilon}_t)\Big]. \tag{A.7}$$

From Eq. (24), it follows that

$$\big(G_{\pi^{\mathrm{NA}}}-G_{\pi_\theta^m}\big)(\mathbf{s}_t,\mathbf{a}_t)=\alpha\,\nabla_{\mathbf{a}_t}\log\pi^h(\mathbf{a}_t\mid\mathbf{s}_t). \tag{A.8}$$

Combining $\|J_f\|_\infty\le C_1$ and $\|\nabla_{\mathbf{a}_t}\log\pi^h\|_\infty\le\kappa(\epsilon)$, we obtain

$$|\Delta_1|\le\alpha\,\|J_f\|_\infty\,\|\nabla_{\mathbf{a}_t}\log\pi^h\|_\infty\le\alpha C_1\,\kappa(\epsilon). \tag{A.9}$$

Since $\|J_f\|_\infty\le C_1$, it follows that

$$|\Delta_2|\le C_1\cdot\Big\|\mathbb{E}_{\mathbf{s}_t}\big[\mathbb{E}_{\mathbf{a}_t\sim\pi^{\mathrm{NA}}}[G_{\pi_\theta^m}]-\mathbb{E}_{\mathbf{a}_t\sim\pi_\theta^m}[G_{\pi_\theta^m}]\big]\Big\|. \tag{A.10}$$

By [43], we have

$$|\Delta_2|\le C_1\,\|G_{\pi_\theta^m}\|_\infty\,\|\pi^{\mathrm{NA}}-\pi_\theta^m\|_1\le C_1(\alpha C_2+C_3)\,\epsilon. \tag{A.11}$$

Combining Eqs. (A.9) and (A.11), and letting $\delta(\epsilon)=\alpha C_1\,\kappa(\epsilon)+C_1(\alpha C_2+C_3)\,\epsilon$, we obtain

$$|\nabla_\theta J(\pi^{\mathrm{NA}})-\nabla_\theta J(\pi_\theta^m)|\le\delta(\epsilon)\to0. \tag{A.12}$$

As $\epsilon\to0$, it follows that $\delta(\epsilon)\to0$. This completes the proof.

References

[1] L. Yang, P. Chi, J. Zhao, and Y. Wang, "Human-in-the-loop formation-containment safe control for multi-agent systems via reinforcement learning," *IEEE Transactions on Artificial Intelligence*, 2025.

[2] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction," *Applied Mechanics Reviews*, vol. 70, no. 1, p. 010804, 2018.

[3] M. Karimi and M. Ahmadi, "iLeAD: An EMG-based adaptive shared control framework for exoskeleton assistance via deep reinforcement learning," *IEEE Transactions on Artificial Intelligence*, vol. 6, no. 10, pp. 2732–2743, 2025.

[4] E. J. Topol, "High-performance medicine: the convergence of human and artificial intelligence," *Nature Medicine*, vol. 25, pp. 44–56, 2019.

[5] X. Yao, J. Liu, X. Zhang, and X. Xu, "Receding-horizon reinforcement learning for time-delayed human–machine shared control of intelligent vehicles," *IEEE Transactions on Human-Machine Systems*, vol. 55, no. 2, pp. 155–164, 2025.

[6] X. Zhuang, D. Li, H. Li, Y. Wang, and J. Zhu, "A dynamic control decision approach for fixed-wing aircraft games via hybrid action reinforcement learning," *Science China Information Sciences*, vol. 68, no. 3, p. 132201, 2025.

[7] W. Wang, X. Na, D. Cao, J. Gong, J. Xi, Y. Xing, and F.-Y. Wang, "Decision-making in driver-automation shared control: A review and perspectives," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 5, pp. 1289–1307, 2020.

[8] T. Nakade, R. Fuchs, H. Bleuler, and J. Schiffmann, "Haptics based multi-level collaborative steering control for automated driving," *Communications Engineering*, vol. 2, no. 1, p. 2, 2023.

[9] H.-N. Wu, X.-Y. Jiang, and M. Wang, "Human cognitive learning in shared control via differential game with bounded rationality and incomplete information," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 10, pp. 5141–5152, 2024.

[10] R. Balachandran, M. De Stefano, H. Mishra, C. Ott, and A. Albu-Schäeffer, "Passive arbitration in adaptive shared control of robots with variable force and stiffness scaling," *Mechatronics*, vol. 90, p. 102930, 2023.

[11] D. Jia, X. Dai, J. Xing, P. Tao, Y. Shi, and Z. Wang, "Asymmetric interaction preference induces cooperation in human–agent hybrid game," *Science China Information Sciences*, vol. 68, no. 11, p. 212201, 2025.

[12] Q. Zhang, P. Li, Y.-B. Zhao, and Y. Kang, "Human-on-the-loop control in surface mount technology via deep reinforcement learning," *IET Control Theory & Applications*, vol. 19, no. 1, p. e70028, 2025.

[13] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[14] R. Luo, M. Zolotas, D. Moore, and T. Padır, "User-customizable shared control for robot teleoperation via Virtual Reality," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2024, pp. 12 196–12 203.

[15] M. Abu-Khalaf, S. Karaman, and D. Rus, "Shared linear quadratic regulation control: A reinforcement learning approach," in *Proc. IEEE 58th Conf. on Decision and Control (CDC)*, 2019, pp. 4569–4576.

[16] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5580–5590.

[17] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 8135–8153, 2023.

[18] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018, pp. 1861–1870.

[19] D. A. Abbink, T. Carlson, M. Mulder, J. C. F. de Winter, F. Aminravan, T. L. Gibo, and E. R. Boer, "A topology of shared control systems: Finding common ground in diversity," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 509–525, 2018.

[20] J. Tan, J. Wang, S. Xue, H. Cao, H. Li, and Z. Guo, "Human–machine shared stabilization control based on safe adaptive dynamic programming with bounded rationality," *International Journal of Robust and Nonlinear Control*, vol. 35, no. 11, pp. 4638–4657, Jul. 2025.

[21] H.-N. Wu and M. Wang, "Learning human behavior in shared control: Adaptive inverse differential game approach," *IEEE Transactions on Cybernetics*, vol. 54, no. 6, pp. 3705–3715, 2024.

[22] J. Tan, S. Xue, Z. Guo, H. Li, H. Cao, and B. Chen, "Data-driven optimal shared control of unmanned aerial vehicles," *Neurocomputing*, vol. 622, p. 129428, 2025.

[23] J. Tan, S. Xue, H. Cao, and S. S. Ge, "Human–ai interactive optimized shared control," *Journal of Automation and Intelligence*, vol. 4, no. 3, pp. 163–176, 2025.

[24] E. Eraslan, Y. Yildiz, and A. M. Annaswamy, "Shared control between pilots and autopilots: An illustration of a cyberphysical human system," *IEEE Control Systems Magazine*, vol. 40, no. 6, pp. 77–97, 2020.

[25] S. Zhao, J. Zhang, R. Zhou, N. Masoud, J. Li, H. Huang, and S. Zhao, "Safety-critical human–machine shared driving for vehicle collision avoidance based on Hamilton–Jacobi reachability," *arXiv preprint arXiv:2502.10610*, 2025.

[26] W. Xu, J. Huang, Y. Wang, C. Tao, and L. Cheng, "Reinforcement learning-based shared control for walking-aid robot and its experimental verification," *Advanced Robotics*, vol. 29, no. 22, pp. 1463–1481, 2015.

[27] S. Reddy, A. D. Dragan, and S. Levine, "Shared autonomy via deep reinforcement learning," in *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

[28] Y. Oh, M. Toussaint, and J. Mainprice, "Learning to arbitrate human and robot control using disagreement between sub-policies," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021, pp. 5305–5311.

[29] C. Schaff and M. R. Walter, "Residual policy learning for shared autonomy," in *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

[30] T. G. J. Rudner, Z. Chen, Y. W. Teh, and Y. Gal, "Tractable function-space variational inference in Bayesian neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022, pp. 22 686–22 698.

[31] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016, pp. 1050–1059.

[32] J. Duan, Y. Guan, S. E. Li, Y. Ren, Q. Sun, and B. Cheng, "Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 11, pp. 6584–6598, 2022.

[33] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[34] S. Singi, Z. He, A. Pan, S. Patel, G. A. Sigurdsson, R. Piramuthu, S. Song, and M. Ciocarlie, "Decision making for human-in-the-loop robotic agents via uncertainty-aware reinforcement learning," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2024, pp. 7939–7945.

[35] M. Krüger, J. Heuer, and T. Schack, "Target uncertainty during motor decision-making: The time course of movement variability reveals the effect of different sources of uncertainty on the control of reaching movements," *Frontiers in Psychology*, vol. 10, p. 41, 2019.

[36] M. Körber, "Theoretical considerations and development of a questionnaire to measure trust in automation," in *Proceedings of the 20th Congress of the International Ergonomics Association (IEA)*, 2018, pp. 13–30.

[37] K. A. Hoff and M. Bashir, "Trust in automation: Integrating empirical evidence on factors that influence trust," *Human Factors*, vol. 57, no. 3, pp. 407–434, 2015.

[38] E. J. de Visser, S. S. Monfort, and F. Krueger, "A little anthropomorphism goes a long way: Effects of oxytocin on trust, compliance, and team performance with automated agents," *Human Factors*, vol. 59, no. 1, pp. 116–130, 2017.

[39] N. Baram, G. Tennenholtz, and S. Mannor, "Maximum entropy reinforcement learning with mixture policies," *arXiv preprint arXiv:2103.10176*, 2021.

[40] S. Ross, G. J. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 627–635.

[41] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6402–6413.

[42] B. Luo, Z. Wu, F. Zhou, and B.-C. Wang, "Human-in-the-loop reinforcement learning in continuous-action space," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 11, pp. 15 735–15 744, 2024.

[43] A. L. Gibbs and F. E. Su, "On choosing and bounding probability metrics," *International Statistical Review*, vol. 70, no. 3, pp. 419–435, 2002.

Shuyue Jiang received the B.S. degree in mathematics and applied mathematics from Anhui Normal University, Wuhu, China, in 2020. She is currently working toward the Ph.D. degree with the University of Science and Technology of China, Hefei, China. Her current research interests include AI-driven human–machine interaction, deep reinforcement learning, embodied intelligence, and large language models for decision-making.

Yun-Bo Zhao received the Ph.D. degree in control theory and control engineering from the University of South Wales, U.K., in 2008. He is currently a Professor with University of Science and Technology of China and also with Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China. His research interests mainly focus on AI-driven control and automation, specifically, human-machine intelligence, smart manufacturing, and AI-driven networked control systems.

Yu Kang received the Ph.D. degree in control theory and control engineering from the University of Science and Technology of China, Hefei, China, in 2005. From 2005 to 2007, he was a Postdoctoral Fellow with the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China. He is currently a Professor with Hefei University of Technology, Hefei, China, and also with the Department of Automation, University of Science and Technology of China, Hefei, China. His current research interests include monitoring of vehicle emissions, adaptive/robust control, variable structure control, mobile manipulator, and Markovian jump systems.

Fei Xie received the B.E. degree from Anhui University of Technology, Ma'ansan, China, in 2018 and the M.E. degree from Hefei University of Technology, Hefei, China, in 2021. He is currently working toward the Ph.D. degree in the Department of Automation at the University of Science and Technology of China, Hefei, China. His research interests include deep learning, fault diagnosis.

Yun-Sheng Zhao received the B.S. degree in automation from the Department of Automation, University of Science and Technology of China (USTC), Hefei, China, in 2021. He is currently pursuing the Ph.D. degree in the same department. His research interests include human–machine collaboration, with a focus on shared control.