SPIKE-BIST: BIDIRECTIONAL STABILIZATION OF SPIKE REPRESENTATION AND SURROGATE GRADIENT FOR MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks provide an energy efficient computing paradigm for decision making by exploiting sparse spike activity and event-driven computation. Training spiking policies under changing conditions remains challenging in multi-agent reinforcement learning. Concurrent policy updates across agents alter observation distributions and learning signals, which can destabilize spike representations and surrogate gradient propagation. To address these issues, we propose Spike-BiST, a bidirectional stabilization framework that coordinates spike representation adaptation and surrogate gradient regulation. A dual-channel adaptive population encoding mechanism is introduced to adapt receptive fields to observations and preserve consistent spike representations across training conditions. Surrogate gradient propagation is regulated by a reliability-driven modulation mechanism that combines reinforcement learning signals with spiking dynamics to reduce gradient fluctuations. Experiments on SMAC, GRF, and MPE demonstrate that Spike-BiST achieves improved task performance and training stability, with lower estimated policy-network inference energy than ANN-based models.

1 INTRODUCTION

Spiking Neural Networks (SNNs) have attracted attention due to event-driven computation, sparse spike representation, and temporal neural dynamics, which provide potential advantages for energy efficient intelligent systems and promote applications in neuromorphic computing Bu et al. (2023), edge intelligence Davies et al. (2021), and autonomous systems Zhu et al. (2024). Unlike conventional artificial neural networks (ANNs) based on continuous activations, SNNs encode and transmit information through binary spike events Zanatta et al. (2024). This computation paradigm reduces the reliance on multiply-accumulate operations and offers potential advantages in inference latency and energy consumption on neuromorphic hardware Roy et al. (2019); Hu et al. (2024).

Extending SNNs to reinforcement learning (RL) introduces challenges in maintaining stable spike representations and surrogate gradient propagation Yao et al. (2023); Yang et al. (2025). Unlike supervised learning with a fixed data distribution, RL collects training data under the current policy, and policy updates change observation distributions and learning signals Liu et al. (2023); Yuan et al. (2019). Shifts in observation distributions can increase the mismatch between fixed encoding parameters and current inputs, leading to variations in spike representations, membrane responses, and firing rates Guo et al. (2021); Zhang et al. (2022); Xiao et al. (2025). The resulting changes in spiking dynamics and learning signals can destabilize surrogate gradient propagation and spiking policy training. Existing methods have explored adaptive spike encoding Tang et al. (2021) and surrogate gradient design Pan et al. (2024) to improve training stability in single-agent settings, but extending these approaches to multi-agent scenarios remains underexplored.

In spiking multi-agent reinforcement learning (MARL), updates to the shared policy alter the behavior of agents, resulting in changes in their local observation distributions and learning signals Saravanan et al. (2021). Unlike single-agent RL, where the learning signal is associated with a single policy, cooperative MARL relies on a shared team objective while individual agents experience different local states and learning signals. This mismatch between shared optimization and agent-specific experience leads to variations in spike representations and optimization across agents and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

training stages. These variations can increase fluctuations in spike responses and surrogate gradient propagation, motivating coordinated adaptation of spike representations and surrogate gradients.

To address these challenges, we propose Spike-BiST, a **bi**directional **st**abilization framework for **spiking** MARL that jointly stabilizes forward spike representations and backward surrogate gradient propagation. For forward representation, Spike-BiST introduces a dual-channel adaptive population encoder to adjust receptive fields according to local observations and agent identity, while structural constraints and temporal smoothing preserve the ordered receptive-field structure. For backward optimization, Spike-BiST employs a reliability-driven surrogate gradient modulation mechanism, which derives a shared surrogate scale from team-level learning progress and performs agent-level refinement using agent-specific advantages and spiking dynamics under critic-confidence regulation.

Our contributions are summarized as follows. 1) We propose a dual-channel adaptive population encoder for spiking MARL that adapts receptive fields to changing observations while preserving structural order and temporal stability, which maintains stable spike representations across training conditions. 2) We introduce a reliability-driven surrogate gradient modulation mechanism that adjusts gradient propagation according to team learning progress, agent-specific learning signals, and spiking dynamics, with critic reliability considered to improve gradient stability. 3) We conduct evaluations on SMAC, GRF, and MPE, demonstrating the generality of Spike-BiST across cooperative scenarios and lower estimated policy-network inference energy than ANN-based models.

2 Related Work

2.1 Spiking Reinforcement Learning

Most existing spiking reinforcement learning (SRL) methods can be grouped into three paradigms: spike-timing-dependent plasticity (STDP)-based learning Asgari et al. (2020); Akl et al. (2022); Kausar et al. (2026), ANN-to-SNN conversion Patela et al. (2019); Hu et al. (2023); Tan et al. (2021), and direct gradient-based training Chen et al. (2022); Li et al. (2024b). STDP-based methods, such as reward-modulated STDP Yang et al. (2025); Yuan et al. (2019), update synaptic weights based on spike timing and reward signals. However, local synaptic updates make credit assignment across deep policy networks challenging Ramezanlou et al. (2025). ANN-to-SNN conversion first trains ANN-based networks and subsequently converts them into spiking counterparts Patela et al. (2019); Li et al. (2021). Discrepancies between ANN activations and spike-based responses degrade policy performance, while accurate conversion often requires multiple internal SNN timesteps and thus increases inference latency Tan et al. (2021). In contrast, direct gradient-based methods use surrogate gradients and spatiotemporal backpropagation to optimize spiking policies under reinforcement learning objectives Kumar et al. (2025); Xiao et al. (2025). This direct optimization paradigm has become the mainstream approach for training spiking policies. In multi-agent settings, related studies have explored SNNs for multi-agent decision making Saravanan et al. (2021) and introduced recurrent spiking architectures for temporal modeling Qin et al. (2025). Despite these advances, the training stability of spiking multi-agent systems has received limited attention. The stability of spike representations and gradient propagation under concurrent policy updates across agents therefore represents a critical issue in spiking multi-agent training.

2.2 Spike Encoding

In SRL, different neural encoding mechanisms map environmental observations to spike representations with distinct trade-offs among sparsity, robustness, and representational capacity Qin et al. (2023). Temporal coding represents inputs through precise spike times and supports sparse computation with low latency Wu et al. (2022); Zhang et al. (2025). Its dependence on exact spike timing makes the representations sensitive to temporal perturbations. Rate coding represents inputs through firing rates accumulated over temporal windows and is less sensitive to individual spike-time variations, although reliable rate estimation requires more internal SNN timesteps Guo et al. (2021). The allocation of each observation dimension across input neurons affects representational capacity. Direct input encoding feeds observations into the first spiking layer without a separate spike-generation stage and reduces encoding overhead Rathi & Roy (2023). However, under a positive firing threshold, negative input values suppress spike generation and therefore cannot produce distinguishable spike responses. To improve representational capacity, dual-neuron encoding assigns separate neu-

rons to positive and negative values and retains sign information in the spike representation Akl et al. (2022; 2023). Population coding further distributes each input across multiple neurons with overlapping tuning curves within a fixed simulation window at the cost of additional neurons Tang et al. (2021); Chen et al. (2024). Population receptive fields can be fixed to preserve structural consistency Zhang et al. (2022) or trainable to adapt to changes in the observation distribution Chen et al. (2025). In general, an effective population encoder needs to preserve structural consistency and adapt to changes in the observation distribution. However, existing population encoding methods have not achieved these two objectives simultaneously.

2.3 Surrogate Gradients

The surrogate gradient method preserves the hard threshold during forward computation and uses an approximate derivative during backpropagation to enable gradient-based optimization of SNNs Zenke & Ganguli (2018); Neftci et al. (2019). Common surrogate profiles include rectangular Wu et al. (2018), triangular Deng et al. (2022), exponential Shrestha & Orchard (2018), and sigmoid-based functions Zenke & Vogels (2021). The width of a surrogate profile determines the range of membrane potentials that receive gradients from the reinforcement learning objective. A broad profile distributes policy gradients across more neuronal states to facilitate exploration early in training, whereas a narrow profile concentrates policy gradients near the firing threshold to promote exploitation later in training. Prior studies adjust surrogate profiles during training through adaptive parameters and schedules Wang et al. (2023); Guo et al. (2024). An adaptive slope schedule Van den Berghe et al. (2025) is introduced to regulate surrogate gradients during reinforcement learning. A trapezoidal surrogate profile Pan et al. (2024) is proposed to broaden the effective gradient region and improve sensitivity to different spiking signal dynamics. Jiang et al. (2025) adapt surrogate parameters to shifts in membrane potential distributions to maintain stable gradient propagation. However, prior work on surrogate gradient adaptation focuses on supervised SNN training or single-agent Spiking RL, with limited applicability to spiking MARL under concurrent policy updates across agents. This gap motivates a surrogate gradient modulation mechanism that combines reinforcement learning signals with spiking dynamics to stabilize policy optimization.

3 Preliminaries

Unlike conventional ANNs, SNNs rely on discrete spike events and temporal neuronal dynamics, which require observation encoding into spike representations and gradient propagation through non-differentiable spiking neurons. At environment step t, agent i receives a local observation $\boldsymbol{o}_i^t \in \mathbb{R}^D$, where D denotes the observation dimension. Population coding represents each observation dimension through P neurons with overlapping Gaussian receptive fields Tang et al. (2021)

$$e_{i,d,p}^t = \exp\left[-(o_{i,d}^t - \mu_{d,p})^2/(2\sigma_{d,p}^2)\right]. \tag{1}$$

where $\mu_{d,p}$ and $\sigma_{d,p} > 0$ denote the center and width of the p-th receptive field for observation dimension d. The responses are concatenated and fed into the spiking policy.

Following standard SNN formulations Chen et al. (2022; 2024), each spiking unit is modeled as a leaky integrate-and-fire (LIF) neuron. At each environment step t, the neuronal states are updated over T_s internal simulation steps indexed by $\tau \in \{1, \ldots, T_s\}$. The membrane integration, spike generation, and soft reset are formulated as

$$\widetilde{V}_t^\tau = \beta V_t^{\tau-1} + I_t^\tau, \quad S_t^\tau = \Theta\left(\widetilde{V}_t^\tau - V_{\mathrm{th}}\right), \quad V_t^\tau = \widetilde{V}_t^\tau - V_{\mathrm{th}} S_t^\tau. \tag{2}$$

where $I_t^\tau \in \mathbb{R}^N$ denotes the input-current vector, $\beta \in (0,1)$ is the membrane leakage factor, and $V_{\mathrm{th}} > 0$ is the firing threshold. The Heaviside step function $\Theta(\cdot)$ is applied element-wise, where $\Theta(x) = 1$ if $x \geq 0$ and $\Theta(x) = 0$ otherwise. Accordingly, $S_t^\tau \in \{0,1\}^N$ contains the binary spike outputs of N neurons. The membrane potential is initialized as $V_t^0 = 0$ at each environment step. The soft-reset mechanism subtracts V_{th} after a spike and preserves the residual membrane potential.

The non-differentiability of the Heaviside step function prevents standard backpropagation Chen et al. (2025). The surrogate gradient method preserves the hard threshold and uses a smooth approximation to its derivative during backpropagation Neftci et al. (2019).

$$\frac{\partial S_t^\tau}{\partial \widetilde{V}_t^\tau} \approx g\left(\widetilde{V}_t^\tau - V_{\mathrm{th}}\right), \quad g(x;k) = \frac{k}{(1+k|x|)^2}, \tag{3}$$

where $g(\cdot)$ is an element-wise surrogate gradient function, and $k > 0$ controls its peak magnitude and concentration around the firing threshold. Gradients are propagated through the neuronal dynamics over the T_s internal simulation steps using backpropagation through time.

4 METHODOLOGY

In this section, we present the Spike-BiST framework, as illustrated in Fig. 1, to mitigate spike representation drift and surrogate gradient instability during spiking MARL training. Specifically, a dual-channel adaptive population encoding mechanism is first introduced to stabilize spike representations during forward computation. Next, a reliability-driven surrogate gradient modulation mechanism is employed to regulate gradient propagation during backpropagation. Finally, the overall training objective is formulated for policy optimization.

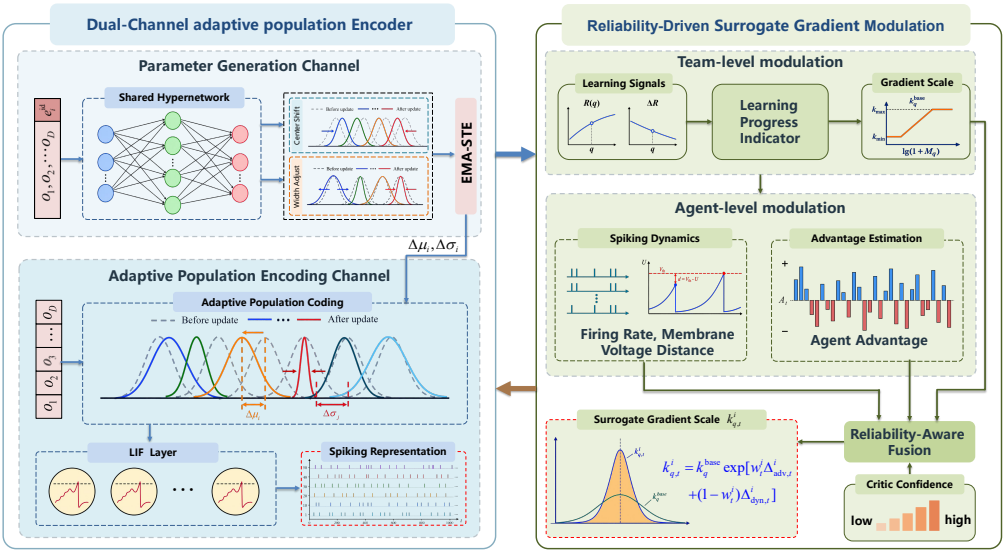

Figure 1: Framework of Spike-BiST: The dual-channel adaptive population encoder provides stable spike representations for policy optimization, while the reliability-driven surrogate gradient modulation provides stable gradient propagation for updating the spiking policy and encoder.

4.1 DUAL-CHANNEL ADAPTIVE POPULATION ENCODING

During spiking MARL training, population encoding needs to accommodate changing observations and preserve receptive-field structure. To meet these requirements, Spike-BiST employs a dual-channel adaptive population encoder with parameter generation and observation encoding channels. The parameter generation channel adapts reference receptive fields, and the observation-encoding channel maps observations through the adapted fields to construct the population representation.

To construct the population representation, each observation dimension is encoded by P neurons with receptive fields. In the parameter generation channel, the receptive-field parameters are adapted through a shared hypernetwork h_ϕ, which generates adaptive adjustments to the centers and widths of the reference receptive fields

$$[\Delta\boldsymbol{\mu}_i^t, \Delta\boldsymbol{\sigma}_i^t] = h_\phi\left([\boldsymbol{o}_i^t; \boldsymbol{e}_i^{\mathrm{id}}]\right), \quad \Delta\boldsymbol{\mu}_i^t, \Delta\boldsymbol{\sigma}_i^t \in \mathbb{R}^{D\times P}, \tag{4}$$

where $\boldsymbol{e}_i^{\mathrm{id}}$ denotes the identity embedding of agent i. The shared hypernetwork h_ϕ avoids maintaining independent parameter generators for individual agents and reduces parameter redundancy. Meanwhile, the identity embedding preserves agent-specific adaptation. Let $\mu_{d,p}^0$ and $\sigma_{d,p}^0 > 0$ denote the reference center and width of population neuron p for observation dimension d, respectively, with the reference centers satisfying $\mu_{d,1}^0 < \cdots < \mu_{d,P}^0$. To adapt these reference receptive

fields and preserve the population structure, the center and width are given as

$$\mu_{i,d,p}^{t} = \mu_{d,p}^{0} + \kappa_d \tanh(\Delta\mu_{i,d,p}^{t}), \quad \sigma_{i,d,p}^{t} = \sigma_{d,p}^{0} \frac{\mathrm{softplus}(\Delta\sigma_{i,d,p}^{t})}{\mathrm{softplus}(0)}, \tag{5}$$

where κ_d controls the maximum center shift for observation dimension d. We set $\kappa_d = \frac{\eta}{2}\min_{r\in\{1,\dots,P-1\}}(\mu_{d,r+1}^{0} - \mu_{d,r}^{0})$, where $\eta \in (0,1)$ is a scaling coefficient. The $\tanh(\cdot)$ function bounds the center shift, and the $\mathrm{softplus}(\cdot)$ function guarantees a positive width and recovers the reference width when $\Delta\sigma_{i,d,p}^{t} = 0$. With the center displacement constrained, the ordering of neighboring receptive fields can be preserved. Since the maximum relative displacement between two adjacent centers is $2\kappa_d$, it follows that

$$\mu_{i,d,p+1}^{t} - \mu_{i,d,p}^{t} \geq \left(\mu_{d,p+1}^{0} - \mu_{d,p}^{0}\right) - 2\kappa_d > 0. \tag{6}$$

Thus, the adapted receptive-field centers remain ordered for every observation, and no pair of centers exchanges positions during adaptation. To account for large variation in the adapted parameters across consecutive environment steps, we use exponential moving-average (EMA) smoothing Rathi & Roy (2023) to reduce temporal variation and employ a straight-through estimator (STE) Zheng et al. (2021) for gradient propagation through the smoothing operation

$$\begin{cases} \widetilde{\mu}_{i,d,p}^{t} = \mu_{i,d,p}^{t} + \mathrm{sg}\left(\gamma_{\mathrm{ema}}\bar{\mu}_{i,d,p}^{t-1} + (1-\gamma_{\mathrm{ema}})\mu_{i,d,p}^{t} - \mu_{i,d,p}^{t}\right), \\ \widetilde{\sigma}_{i,d,p}^{t} = \sigma_{i,d,p}^{t} + \mathrm{sg}\left(\gamma_{\mathrm{ema}}\bar{\sigma}_{i,d,p}^{t-1} + (1-\gamma_{\mathrm{ema}})\sigma_{i,d,p}^{t} - \sigma_{i,d,p}^{t}\right), \end{cases} \tag{7}$$

where $\gamma_{\mathrm{ema}} \in [0,1)$ denotes the smoothing coefficient, $\bar{\mu}_{i,d,p}^{t-1}$ and $\bar{\sigma}_{i,d,p}^{t-1}$ denote the EMA-smoothed center and width retained from the previous environment step, and $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. Note that the EMA-smoothed receptive-field parameters are used in forward computation to reduce fluctuations in the hypernetwork outputs and stabilize the receptive fields. During back-propagation, gradients are computed with respect to the receptive-field parameters and propagated to the hypernetwork. This gradient path avoids attenuation caused by EMA smoothing.

Moreover, the observation-encoding channel uses the adapted and smoothed receptive-field parameters to construct the population representation. Each neuron is characterized by the smoothed center $\widetilde{\mu}_{i,d,p}^{t}$ and width $\widetilde{\sigma}_{i,d,p}^{t}$. The response of neuron p for observation dimension d is given by

$$\widetilde{e}_{i,d,p}^{t} = \exp\left[-\frac{1}{2}\left((o_{i,d}^{t} - \widetilde{\mu}_{i,d,p}^{t})/(\widetilde{\sigma}_{i,d,p}^{t} + \epsilon)\right)^2\right] \tag{8}$$

where $\epsilon > 0$ is a small constant for numerical stability. The population responses form a matrix $\widetilde{\boldsymbol{e}}_i^{t} \in \mathbb{R}^{D\times P}$, which is vectorized as $\boldsymbol{x}_i^{t} = \mathrm{vec}(\widetilde{\boldsymbol{e}}_i^{t}) \in \mathbb{R}^{DP}$ and subsequently fed into the LIF policy network. Overall, the dual-channel encoder uses structurally constrained and temporally smoothed receptive-field adaptation to construct stable population representations and mitigate spike representation drift during spiking MARL training.

4.2 Reliability-Driven Surrogate Gradient Modulation

Surrogate gradient adaptation remains challenging in spiking MARL, where concurrent policy updates across agents amplify gradient instability across agents. The proposed Spike-BiST addresses this challenge through a reliability-driven surrogate gradient modulation mechanism with team-level and agent-level modulation. The team-level modulation derives a shared base scale from team learning signals and the agent-level modulation combines advantage-based and dynamics-based corrections under a critic-confidence gate to determine the final surrogate gradient scale for each agent.

The shared base surrogate scale is defined as

$$k_q^{\mathrm{base}} = \mathrm{clip}\left(k_0 + \kappa_R \log\left(1 + M_q\right), k_0^{\min}, k_0^{\max}\right), \tag{9}$$

where q denotes the surrogate gradient scale update index, k_0 is the initial surrogate scale, κ_R controls the influence of the team-level learning progress indicator, and $k_0^{\min}$ and $k_0^{\max}$ denote the lower and upper bounds of the base scale. The learning progress indicator M_q is defined as

$$M_q = \mathrm{EMA}_M\left(\frac{[\bar{\mathcal{R}}_q]_+ + \eta_R[\Delta\bar{\mathcal{R}}_q]_+}{1 + \mathrm{EMA}_V\left(|\Delta\bar{\mathcal{R}}_q|\right)}\right), \tag{10}$$

where $\bar{\mathcal{R}}_q$ is the normalized mean team return and $\Delta\bar{\mathcal{R}}_q$ is its change from the previous scale update. The coefficient η_R weights positive return changes through $[x]_+ = \max(x, 0)$, and the EMA operators suppress short-term fluctuations.

According to Eqs. 9 and 10, low team returns and large return variations during early training yield a small M_q and keep k_q^{base} close to k_0. Under Eq. 3, this scale produces a broad surrogate profile that allows more spiking neurons to receive policy gradients and supports exploration. As team returns improve and stabilize, M_q and k_q^{base} increase, and the profile narrows around the firing threshold to favor exploitation through selective policy updates. This design adapts the surrogate gradient scale to learning progress, aligns the gradient profile with the transition from exploration to exploitation, and improves gradient propagation stability.

Unlike single-agent RL, where the individual return can reflect the learning progress of the policy, the shared return in MARL provides a team-level learning signal and cannot capture agent-specific optimization progress, which in general leads to inappropriate surrogate gradient scaling. Therefore, to account for agent-level differences and reduce surrogate gradient instability under concurrent policy updates, we refine the shared base scale using agent-specific advantage estimates and spiking dynamics. Specifically, for the advantage-based correction, estimates are normalized within each agent type group to account for differences in advantage magnitude.

$$z_t^i = (\hat{A}_t^i - \mu_{A,q}^{g_i})/(\sigma_{A,q}^{g_i} + \epsilon), \tag{11}$$

where $\hat{A}_t^i$ denotes the agent-specific advantage estimate, g_i denotes the type group of agent i, and $\mu_{A,q}^{g_i}$ and $\sigma_{A,q}^{g_i}$ denote the mean and standard deviation of the advantage estimates within that group, respectively. Based on the normalized advantage, the correction is defined as

$$\Delta_{\text{adv},t}^i = \frac{a_0 \tanh(\eta_A z_t^i)}{1 + \lambda_A \left[k_q^{\text{base}}/k_0 - 1\right]_+}, \tag{12}$$

where a_0 controls the correction magnitude, η_A determines its sensitivity to the normalized advantage, and λ_A controls its attenuation. In parallel, we consider the effect of spiking dynamics on surrogate gradient stability and introduce a dynamics-based correction by using the firing rate and the normalized absolute membrane-threshold distance

$$\rho_t^i = \frac{1}{T_s|\mathcal{M}|}\sum_{\tau=1}^{T_s}\sum_{m\in\mathcal{M}} S_{t,\tau}^{i,m}, \quad d_t^i = \frac{1}{T_s|\mathcal{M}|}\sum_{\tau=1}^{T_s}\sum_{m\in\mathcal{M}} \frac{|\widetilde{V}_{t,\tau}^{i,m} - V_{\text{th}}|}{V_{\text{th}} + \epsilon}, \tag{13}$$

where $\mathcal{M}$ denotes the neuron set used to compute the spiking statistics. The normalized absolute membrane-threshold distance affects the surrogate derivative, and the firing rate indicates silent or saturated activity. These statistics jointly characterize the spiking dynamics and are mapped to

$$\Delta_{\text{dyn},t}^i = a_1 \tanh\left(G_{\text{dyn}}(\text{sg}([\rho_t^i, d_t^i]^\top))\right), \tag{14}$$

where a_1 bounds the correction magnitude, and G_{dyn} is a lightweight network. Since the advantage-based correction depends on critic estimates, a critic-confidence gate derived from the relative value prediction loss regulates its contribution

$$g_{V,t}^i = \text{clip}\left(\exp\left(-\beta_V\left[\frac{\ell_{V,t}^i}{\bar{\ell}_V^{q-1} + \epsilon} - 1\right]_+\right), g_{\min}, 1\right), \tag{15}$$

where $\ell_{V,t}^i$ is the per-sample value prediction loss and $\bar{\ell}_V^{q-1}$ is its historical mean before the current scale update. The parameter β_V controls sensitivity to relative value loss, and $g_{\min}$ sets the gate lower bound. When the relative value loss exceeds its historical reference, $g_{V,t}^i$ decreases with the excess and limits the advantage-based contribution. This confidence estimate is then incorporated into the adaptive fusion weight as follows

$$w_t^i = g_{V,t}^i \,\text{sigmoid}\left(G_{\text{gate}}\left(\text{sg}\left([\rho_t^i, d_t^i, \boldsymbol{e}_i^{\text{id}}]^\top\right)\right)\right), \tag{16}$$

where G_{gate} is a lightweight fusion network. Finally, the two corrections are fused according to w_t^i and applied to the shared base scale to obtain the final agent-specific surrogate gradient scale

$$k_{q,t}^i = \text{clip}\left(k_q^{\text{base}} \exp\left[(1 - w_t^i)\Delta_{\text{dyn},t}^i + w_t^i \Delta_{\text{adv},t}^i\right], k^{\min}, k^{\max}\right), \tag{17}$$

where $k^{\min}$ and $k^{\max}$ denote the lower and upper bounds, respectively. Overall, the team-level modulation aligns the surrogate profile with the current learning stage to stabilize surrogate gradient approximation. The agent-level modulation then assigns agent-specific scales to mitigate the amplification of gradient fluctuations during concurrent policy updates across agents.

4.3 Optimization Objective

Given that surrogate gradient modulation requires agent-specific advantage estimates, the proposed method is implemented within the MAPPO framework Yu et al. (2022). To further stabilize receptive-field adaptation, surrogate scale modulation, and spike activity, three auxiliary regularizers are added to the policy objective.

The first regularizer controls receptive-field widths in the adaptive population encoder. Although the normalized softplus parameterization in Eq. 5 ensures positive widths, it does not restrict them to the prescribed interval $[\sigma_{\min}, \sigma_{\max}]$. Deviations from this interval are penalized as

$$\mathcal{L}_\sigma = \frac{1}{|\mathcal{V}_q| DP} \sum_{(i,t)\in\mathcal{V}_q} \sum_{d=1}^{D} \sum_{p=1}^{P} \left(\left[\log \frac{\sigma_{\min}}{\sigma_{i,d,p}^t + \epsilon} \right]_+^2 + \left[\log \frac{\sigma_{i,d,p}^t + \epsilon}{\sigma_{\max}} \right]_+^2 \right), \tag{18}$$

where $\mathcal{V}_q$ denotes the set of valid agent-step samples used to evaluate the auxiliary regularizers. The second regularizer penalizes abrupt changes in surrogate gradient scales between consecutive environment steps and excessive deviations from the shared base scale

$$\mathcal{L}_{\mathrm{SGM}} = \frac{1}{|\mathcal{C}_q|} \sum_{(i,t)\in\mathcal{C}_q} (k_{q,t}^i - k_{q,t-1}^i)^2 + \frac{\lambda_b}{|\mathcal{V}_q|} \sum_{(i,t)\in\mathcal{V}_q} (k_{q,t}^i - k_q^{\mathrm{base}})^2, \tag{19}$$

where $\mathcal{C}_q$ indexes valid consecutive sample pairs for the same agent within an episode, and $\lambda_b \geq 0$ controls the penalty on deviations from the base scale. The third regularizer discourages excessively sparse or saturated spike activity through a penalty on firing rates outside the target interval

$$\mathcal{L}_{\mathrm{fr}} = \frac{1}{|\mathcal{V}_q|} \sum_{(i,t)\in\mathcal{V}_q} \left([\rho_{\min} - \rho_t^i]_+^2 + [\rho_t^i - \rho_{\max}]_+^2 \right), \tag{20}$$

where ρ_t^i is computed according to Eq. 13. The penalty vanishes when the firing rate lies within the target interval. We combine these regularizers with the MAPPO objective to obtain

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_\pi + c_V \mathcal{L}_V - c_H \mathcal{H}_\pi + \lambda_\sigma \mathcal{L}_\sigma + \lambda_{\mathrm{SGM}} \mathcal{L}_{\mathrm{SGM}} + \lambda_\rho \mathcal{L}_{\mathrm{fr}}, \tag{21}$$

where $\mathcal{L}_\pi$, $\mathcal{L}_V$, and $\mathcal{H}_\pi$ represent the standard clipped policy loss, centralized value loss, and policy entropy, respectively. The coefficients c_V and c_H weight the value loss and entropy term, λ_σ, λ_{SGM}, and λ_ρ control the contributions of the three auxiliary regularizers. The objective couples policy and value-function optimization with explicit regularization of receptive-field widths, surrogate gradient scales, and spike activity. The pseudocode for the above learning process is given in Algorithm 1.

5 Experiments

In this section, we conduct experiments to answer the following three questions **Q1-Q3**. **Q1.** How does Spike-BiST compare with SOTA MARL and SNN-based MARL methods in cooperative multi-agent tasks? **Q2.** How does each proposed component contribute to the performance of Spike-BiST? **Q3.** How does Spike-BiST affect energy efficiency compared with ANN-based policies?

We conduct experiments on SMAC Samvelyan et al. (2019), GRF Kurach et al. (2020), and MPE Lowe et al. (2017) benchmarks to evaluate Spike-BiST on cooperative multi-agent tasks. The comparisons include spiking MARL methods (GRSN Qin et al. (2025)), MARL methods (EMU Na et al. (2024) and ICES Li et al. (2024a)), and SNN-based policy methods (SNN-MAPPO Yu et al. (2022) and ILC-SAN Chen et al. (2025)). Detailed descriptions of the environments, experimental settings, and implementation details are provided in Appendices A.2.1-A.2.3, respectively.

5.1 **Q1.** Performance Comparison on Multi-Agent Benchmarks

To evaluate Spike-BiST across different cooperative settings, we compare it with existing methods on representative tasks from SMAC, GRF, and MPE. Figure 2 presents the learning curves, while detailed experimental settings and additional results are given in Appendices A.2.3 and A.3.1.

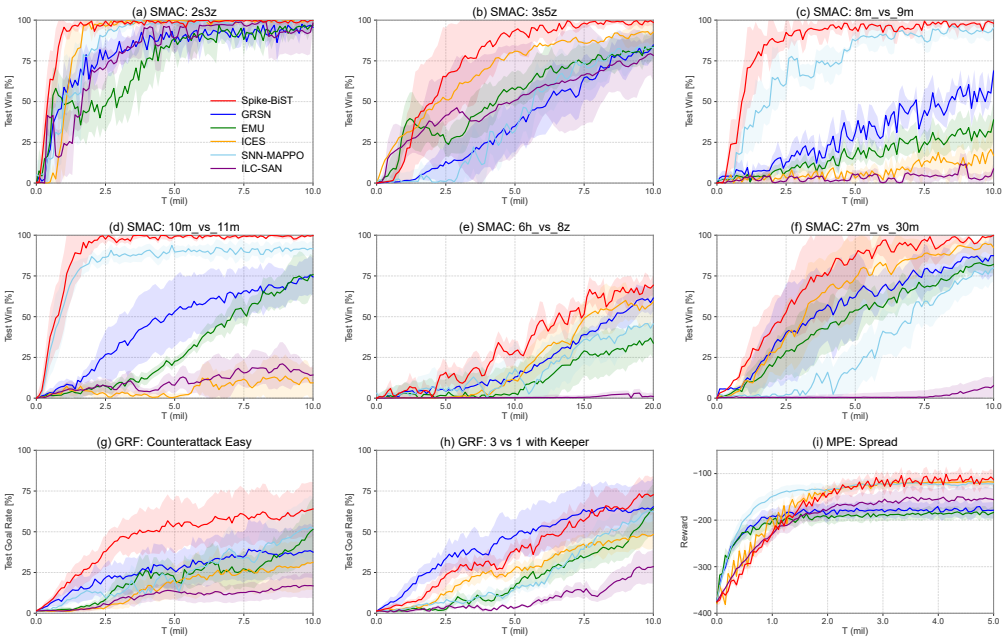

Figure 2: Performance comparison of Spike-BiST and baseline methods across multiple cooperative multi-agent tasks: (a) 2s3z; (b) 3s5z; (c) 8m_vs_9m; (d) 10m_vs_11m; (e) 6h_vs_8z; (f) 27m_vs_30m; (g) Counterattack Easy; (h) 3 vs 1 with Keeper; and (i) Spread.

Spike-BiST achieves better performance across most evaluated scenarios, with faster convergence and smaller performance fluctuations than existing methods, especially in tasks with higher coordination requirements. These results indicate that the proposed bidirectional stabilization mechanism improves the optimization stability of spiking policies. The consistent gains across SMAC, GRF, and MPE suggest that Spike-BiST remains effective under different cooperative scenarios. Moreover, the reduced performance fluctuations reflect more stable policy updates throughout training.

5.2 **Q2.** Ablation Studies

To investigate the contribution of different components in Spike-BiST, ablation studies are conducted on the 6h_vs_8z task of the SMAC benchmark. The effectiveness of the dual-channel adaptive population encoding and the reliability-driven surrogate gradient modulation is first evaluated. Then, the design choices of the adaptive encoding mechanism and the modulation components are further analyzed. As shown in Fig. 3, Spike-BiST is compared with variants that remove specific components or replace adaptive mechanisms with fixed alternatives.

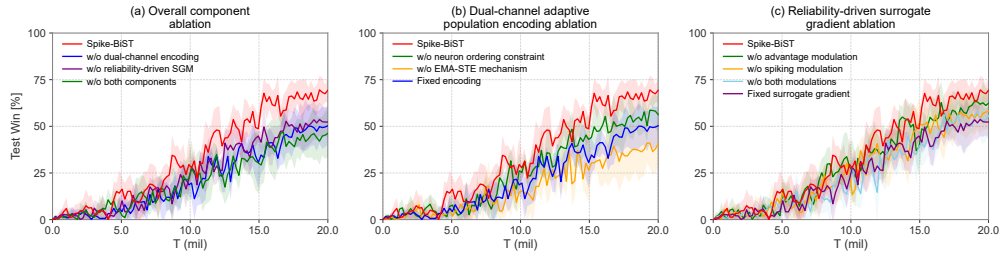

Figure 3: Ablation results of Spike-BiST on SMAC: (a) Overall component ablation; (b) Dual-channel adaptive population encoding ablation; (c) Reliability-driven surrogate gradient ablation.

As shown in Fig. 3(a), Spike-BiST achieves a test win rate of 69.3%, higher than the variants with one proposed component and the version with fixed population encoding and surrogate gradients. The complete model achieves the best performance and confirms the complementary effects

of dual-channel adaptive population encoding and reliability-driven surrogate gradient modulation. Figure 3(b) evaluates adaptive population encoding. The variant without neuron ordering constraints still outperforms fixed encoding but remains below Spike-BiST, which demonstrates the importance of receptive-field adaptation and structural consistency. The absence of EMA-STE causes a further performance drop and suggests that EMA-STE stabilizes receptive-field updates and gradient propagation. Figure 3(c) evaluates surrogate gradient modulation. The variants without advantage-based or dynamics-based modulation both achieve lower test win rates, and the performance further declines when both terms are absent or a fixed surrogate gradient replaces the adaptive one. These results confirm that reinforcement learning signals and spiking dynamics both contribute to surrogate gradient adaptation. Additional experiments on MuJoCo tasks are provided in Appendix A.3.2 to evaluate the effectiveness of the proposed adaptive population encoding in control tasks.

5.3 **Q3.** Energy Efficiency Analysis

To evaluate the inference energy efficiency of Spike-BiST, we estimate the computational cost of the actor by counting synaptic operations (SynOps), following previous SNN studies Qin et al. (2023). The SynOps of SNNs and ANNs are calculated as

$$\text{SynOps} = \sum_{t=1}^{T} \sum_{l=1}^{L-1} \sum_{i=1}^{N_l} f_{\text{out}}^{l} s_i^l[t] \text{ (SNN)}, \quad \text{SynOps} = \sum_{l=1}^{L} f_{\text{in}}^{l} N_l \text{ (ANN)}, \tag{22}$$

where N_l is the number of neurons in layer l, and f_{out}^{l} and f_{in}^{l} denote the numbers of outgoing and incoming connections, respectively. We randomly sample 1,024 inference inputs to estimate the average firing activity and SynOps. Following Qin et al. (2023), we assume 0.9 pJ for a spike-driven SOP and 4.6 pJ for a MAC operation, and show the resulting energy consumption in Table 1.

Table 1: Synaptic operations and estimated inference energy of the policy network.

Metric	SMAC 3m		GRF counterattack (easy)		MPE spread	
	ANN	Spike-BiST	ANN	Spike-BiST	ANN	Spike-BiST
Dense MACs	149.8K	2.3K	143.6K	4.9K	137.0K	1.3K
Spike SOPs	–	382.2K	–	212.4K	–	140.4K
Estimated Energy (μJ)	0.6889	0.3546	0.6606	0.2135	0.6300	0.1322
Energy Saved	**48.53%**		**67.68%**		**79.01%**	

As shown in Table 1, Spike-BiST reduces the estimated policy-network inference energy by 48.53%, 67.68%, and 79.01% on SMAC 3m, GRF counterattack (easy), and MPE spread, respectively. The reduction results from the use of a small number of dense MACs and a larger proportion of low-cost spike-driven SOPs. These results demonstrate the energy-efficiency advantage of the spiking policy computation.

6 Conclusion

In this paper, we propose Spike-BiST to mitigate spike representation drift and surrogate gradient instability in spiking multi-agent reinforcement learning. A dual-channel adaptive population encoder is introduced to adjust receptive-field centers and widths under structural and temporal constraints, thus stabilizing spike representations. The reliability-driven surrogate gradient modulation mechanism derives a shared base scale from team-level learning progress. Building on this scale, agent-specific adjustments integrate advantage signals and spiking dynamics through a critic-confidence gate to regulate gradient propagation. Experiments demonstrate the effectiveness of Spike-BiST in cooperative multi-agent tasks, while ablation studies verify the contributions of the proposed components. Spike-BiST also retains the energy-efficiency advantage of spike-driven computation with lower estimated policy-network inference energy than its ANN counterpart. These results support the use of coordinated representation and gradient adaptation in spiking policy optimization.

Ethics Statement

Our submission follows the ICLR Code of Ethics. We do not identify any specific ethical concerns in this work.

Reproducibility Statement

Detailed implementation settings, network architectures, training procedures, and hyperparameters are provided in the Appendix. All experiments were conducted using two NVIDIA RTX 4090 GPUs. The experimental configurations and evaluation protocols are described to facilitate reproduction and independent verification of the reported results.

Use of Large Language Models

Large Language Models (LLMs) were employed only to improve the linguistic quality and readability of the manuscript, including grammar correction and minor wording refinement. The research ideas, theoretical derivations, algorithmic development, experimental design, and implementation were completed independently without relying on LLMs.

References

Mahmoud Akl, Yulia Sandamirskaya, Deniz Ergene, Florian Walter, and Alois Knoll. Fine-tuning deep reinforcement learning policies with R-STDP for domain adaptation. In *5th International Conference on Neuromorphic Systems*, pp. 1–8, Knoxville, USA, July 2022.

Mahmoud Akl, Deniz Ergene, Florian Walter, and Alois Knoll. Toward robust and scalable deep spiking reinforcement learning. *Frontiers in Neurorobotics*, 16, 2023. Art. no. 1075647.

Hajar Asgari, Mazloom-Nezhad Maybodi Babak, Payvand Melika, and Rahimi Azghadi Mostafa. Low-energy and fast spiking neural network for context-dependent learning on FPGA. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(11):2697–2701, 2020.

Tong Bu, Jianhao Ding, Zecheng Hao, and Zhaofei Yu. Rate gradient approximation attack threats deep spiking neural networks. In *34th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7896–7906, Vancouver, Canada, June 2023.

Ding Chen, Peixi Peng, Tiejun Huang, and Yonghong Tian. Deep reinforcement learning with spiking Q-learning. *arXiv preprint arXiv:2201.09754*, 2022.

Ding Chen, Peixi Peng, Tiejun Huang, and Yonghong Tian. Fully spiking actor network with intralayer connections for reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):2881–2893, 2025.

Xinyi Chen, Qu Yang, Jibin Wu, Haizhou Li, and Kay Chen Tan. A hybrid neural coding approach for pattern recognition with spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3064–3078, 2024.

Mike Davies, Wild Andreas, Orchard Garrick, Sandamirskaya Yulia, A. Fonseca Guerra Gabriel, Joshi Prasad, and R. Risbud Sumedh. Advancing neuromorphic computing with Loihi: A survey of results and outlook. *Proceedings of the IEEE*, 109(5):911–934, 2021.

Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. *arXiv preprint arXiv:2202.11946*, 2022.

Wenzhe Guo, Mohammed E. Fouda, Ahmed M. Eltawil, and Khaled Nabil Salama. Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems. *Frontiers in Neuroscience*, 15, 2021. Art. no. 638474.

Yufei Guo, Yuanpei Chen, Zecheng Hao, Weihang Peng, Jie Zhou, Yuhan Zhang, Xiaode Liu, and Zhe Ma. Take a shortcut back: Mitigating the gradient vanishing for training spiking neural networks. In *38th International Conference on Neural Information Processing Systems*, pp. 24849–24867, Vancouver, Canada, December 2024.

Yangfan Hu, Qian Zheng, Xudong Jiang, and Gang Pan. Fast-SNN: Fast spiking neural network by converting quantized ANN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (12):14546–14562, 2023.

Yangfan Hu, Qian Zheng, Guoqi Li, Huajin Tang, and Gang Pan. Toward large-scale spiking neural networks: A comprehensive survey and future directions. *arXiv preprint arXiv:2409.02111*, 2024.

Jiaqiang Jiang, Lei Wang, Runhao Jiang, Jing Fan, and Rui Yan. Adaptive gradient learning for spiking neural networks by exploiting membrane potential dynamics. *arXiv:2505.11863*, 2025.

Rizwana Kausar, Sudevan Vidya, Viegas Jaime, and Dias Jorge. A two-stage biologically inspired robot navigation framework via reward-modulated STDP and obstacle-state encoding. *IEEE Robotics and Automation Letters*, 11(4):4681–4688, 2026.

Aakash Kumar, Lei Zhang, Hazrat Bilal, Shifeng Wang, Shaikh Ali Muhammad, Lu Bo, Avinash Rohra, and Alisha Khalid. DSQN: Robust path planning of mobile robot based on deep spiking Q-network. *Neurocomputing*, 634, 2025. Art. no. 129916.

Karol Kurach, Anton Raichuk, Piotr Stańczyk, and Michał Zajac. Google research football: A novel reinforcement learning environment. In *34th AAAI Conference on Artificial Intelligence*, pp. 4501–4510, New York, USA, February 2020.

Xinran Li, Zifan Liu, Shibo Chen, and Jun Zhang. Individual contributions as intrinsic exploration scaffolds for multi-agent reinforcement learning. In *41st International Conference on Machine Learning*, pp. 28387–28402, Vienna, Austria, July 2024a.

Yang Li, Feifei Zhao, Dongcheng Zhao, and Yi Zeng. Directly training temporal spiking neural network with sparse surrogate gradient. *Neural Networks*, 179, 2024b. Art. no. 106499.

Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ANN: Towards efficient, accurate spiking neural networks calibration. In *38th International Conference on Machine Learning*, pp. 6316–6325, Online, July 2021.

Guisong Liu, Wenjie Deng, Xiurui Xie, Li Huang, and Huajin Tang. Human-level control through directly trained deep spiking Q-networks. *IEEE Transactions on Cybernetics*, 53(11):7187–7198, 2023.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *31st International Conference on Neural Information Processing Systems*, pp. 6382–6393, Long Beach, USA, December 2017.

Hyungho Na, Yunkyeong Seo, and Il-chul Moon. Efficient episodic memory utilization of cooperative multi-agent reinforcement learning. In *12th International Conference on Learning Representations*, pp. 47821–47853, Vienna, Austria, May 2024.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

Yuhao Pan, Xiucheng Wang, Nan Cheng, and Qi Qiu. Trapezoidal gradient descent for effective reinforcement learning in spiking networks. In *2024 International Conference on Ubiquitous Communication*, pp. 192–196, Xi'an, China, July 2024.

Devdhar Patela, Hazan Hananel, J. Saunders Daniel, T. Siegelmann Hava, and Kozma Robert. Improved robustness of reinforcement learning policies upon conversion to spiking neuronal network platforms applied to Atari Breakout game. *Neural Networks*, 120:108–115, 2019.

Lang Qin, Rui Yan, and Huajin Tang. A low latency adaptive coding spike framework for deep reinforcement learning. In *32nd International Joint Conference on Artificial Intelligence*, pp. 3049–3057, Macau, China, August 2023.

Lang Qin, Ziming Wang, Runhao Jiang, Rui Yan, and Huajin Tang. GRSN: Gated recurrent spiking neurons for POMDPs and MARL. In *39th AAAI Conference on Artificial Intelligence*, pp. 1483–1491, Philadelphia, USA, February 2025.

Mohammad Tayefe Ramezanlou, Howard Schwartz, Ioannis Lambadaris, and Michel Barbeau. Enhancing cooperative multi-agent reinforcement learning through the integration of R-STDP and federated learning. *Neurocomputing*, 617, 2025. Art. no. 129005.

Nitin Rathi and Kaushik Roy. Diet-snn: Direct input encoding with leakage and threshold optimization in deep spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(6):3174–3182, 2023.

Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. In *18th International Conference on Autonomous Agents and Multiagent Systems*, pp. 2186–2188, Montreal, Canada, May 2019.

M. Saravanan, Satheesh Kumar Perepu, Kaushik Dey, Sreeja Gaddamidi, and Adhesh Reghu Kumar. Exploring spiking neural networks in single and multi-agent RL methods. In *2021 International Conference on Rebooting Computing*, pp. 88–98, Online, November 2021.

Sumit Bam Shrestha and Garrick Orchard. SLAYER: Spike layer error reassignment in time. In *32nd International Conference on Neural Information Processing Systems*, pp. 1419–1428, Montréal, Canada, December 2018.

Weihao Tan, Devdhar Patel, and Robert Kozma. Strategy and benchmark for converting deep Q-networks to event-driven spiking neural networks. In *35th AAAI Conference on Artificial Intelligence*, pp. 9816–9824, Vancouver, Canada, February 2021.

Guangzhi Tang, Neelesh Kumar, Raymond Yoo, and Konstantinos P. Michmizos. Deep reinforcement learning with population-coded spiking neural network for continuous control. In *4th Conference on Robot Learning*, pp. 2016–2029, London, UK, November 2021.

Korneel Van den Berghe, Stein Stroobants, Vijay Janapa Reddi, and G. C. H. E. de Croon. Adaptive surrogate gradients for sequential reinforcement learning in spiking neural networks. In *39th Conference on Neural Information Processing Systems*, pp. 147904–147926, San Diego, USA, December 2025.

Ziming Wang, Runhao Jiang, Shuang Lian, Rui Yan, and Huajin Tang. Adaptive smoothing gradient learning for spiking neural networks. In *40th International Conference on Machine Learning*, pp. 35798–35816, Honolulu, USA, July 2023.

Guanlin Wu, Dongchen Liang, Shaotong Luan, and Ji Wang. Training spiking neural networks for reinforcement learning tasks with temporal coding method. *Frontiers in Neuroscience*, 16, 2022. Art. no. 877701.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, 12, 2018. Art. no. 231.

Rong Xiao, Jie Zhang, Chenwei Tang, and Jiancheng Lv. Multi-attribute dynamic attenuation learning improved spiking actor network. *Neurocomputing*, 614, 2025. Art. no. 128819.

Zhile Yang, Shangqi Guo, Ying Fang, Zhaofei Yu, and Jian K. Liu. Spiking variational policy gradient for brain inspired reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):1975–1990, 2025.

Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. In *37th International Conference on Neural Information Processing Systems*, pp. 64043–64058, New Orleans, USA, December 2023.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. In *36th Conference on Neural Information Processing Systems*, pp. 24611–24624, New Orleans, USA, December 2022.

Mengwen Yuan, Xi Wu, Rui Yan, and Huajin Tang. Reinforcement learning in spiking neural networks with stochastic and deterministic synapses. *Neural Computation*, 31(12):2368–2389, 2019.

Luca Zanatta, Francesco Barchi, Simone Manoni, Silvia Tolu, Andrea Bartolini, and Andrea Acquaviva. Exploring spiking neural networks for deep reinforcement learning in robotic tasks. *Scientific Reports*, 14(1), 2024. Art. no. 30648.

Friedemann Zenke and Surya Ganguli. SuperSpike: Supervised learning in multilayer spiking neural networks. *Neural Computation*, 30(6):1514–1541, 2018.

Friedemann Zenke and Tim P. Vogels. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural Computation*, 33(4):899–925, 2021.

Duzhen Zhang, Tielin Zhang, Shuncheng Jia, and Bo Xu. Multi-scale dynamic coding improved spiking actor network for reinforcement learning. In *36th AAAI Conference on Artificial Intelligence*, pp. 59–67, Washington, USA, February 2022.

Malu Zhang, Shuai Wang, Jibin Wu, Wenjie Wei, Dehao Zhang, Zijian Zhou, Siying Wang, Fan Zhang, and Yang Yang. Toward energy-efficient spike-based deep reinforcement learning with temporal coding. *IEEE Computational Intelligence Magazine*, 20(2):45–57, 2025.

Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *35th AAAI Conference on Artificial Intelligence*, pp. 11062–11070, Vancouver, Canada, February 2021.

Rui-Jie Zhu, Ziqing Wang, Leilani Gilpin, and Jason K. Eshraghian. Autonomous driving with spiking neural networks. In *38th International Conference on Neural Information Processing Systems*, pp. 136782–136804, Vancouver, Canada, December 2024.

A Appendix

A.1 Architecture and Training Procedure

The adaptive encoder and surrogate-gradient modulation use small shared MLPs. For the parameter generation channel, the hypernetwork h_ϕ has the structure

$$\mathrm{Linear}(D + d_{\mathrm{id}}, 64) \to \mathrm{ReLU} \to \mathrm{Linear}(64, 2DP),$$

where $d_{\mathrm{id}} = 1$ is the dimension of the trainable agent-identity embedding. The output is reshaped into the center and width offsets $[\Delta\boldsymbol{\mu}, \Delta\boldsymbol{\sigma}] \in \mathbb{R}^{2\times D\times P}$. The same h_ϕ is shared by all agents, and the identity embedding distinguishes agent-specific inputs. For observation dimension d, the P reference centers are uniformly initialized over $[o_{\min,d}, o_{\max,d}]$, with spacing

$$\delta_{\mu,d} = \frac{o_{\max,d} - o_{\min,d}}{P - 1},$$

and the reference width is set to $\sigma^0_{d,p} = 1.2\delta_{\mu,d}$. We use $\eta = 0.8$ for the center-shift coefficient and 0.99 for EMA smoothing. The dynamics network G_{dyn} follows

$$\mathrm{Linear}(2, 32) \to \mathrm{ReLU} \to \mathrm{Linear}(32, 1),$$

with $[\rho^i_t, d^i_t]$ as input. The scalar output is transformed by $\tanh$ and scaled by a_1 to obtain $\Delta^i_{\mathrm{dyn},t}$. The fusion network G_{gate} follows

$$\mathrm{Linear}(2 + d_{\mathrm{id}}, 32) \to \mathrm{ReLU} \to \mathrm{Linear}(32, 1),$$

and receives $[\rho^i_t, d^i_t, \boldsymbol{e}^{\mathrm{id}}_i]$. The output is passed through a sigmoid and combined with the critic-confidence term to obtain w^i_t. Both G_{dyn} and G_{gate} share parameters across agents. Algorithm 1 summarizes the training procedure.

Algorithm 1 Training procedure of Spike-BiST

Require: Rollout length H, PPO epochs E, and SNN steps T_s
Ensure: Trained decentralized spiking policy (θ, ϕ)

Initialize policy θ, encoder ϕ, critic ψ, modulation networks ω, receptive fields, and EMA states
Initialize surrogate-scale update index $q \leftarrow 0$
for training iteration $r = 1, 2, \ldots$ **do**
  Set behavior parameters $(\theta_{\text{old}}, \phi_{\text{old}}) \leftarrow (\theta, \phi)$
  Collect an H-step rollout and estimate return $\hat{R}_t^i$ and advantage $\hat{A}_t^i$
  for PPO epoch $e = 1, \ldots, E$ **do**
    for temporally ordered sequence minibatch $\mathcal{B}_m$ **do**
      $q \leftarrow q + 1$
      Compute current team-return statistics and obtain M_q and k_q^{base}
      Normalize advantages within each agent type to obtain z_t^i
      Adapt receptive fields and construct population representations
      Evaluate the spiking policy over T_s internal SNN steps and evaluate the centralized critic
      Compute spiking statistics ρ_t^i and d_t^i
      Derive advantage- and dynamics-based corrections $\Delta_{\text{adv},t}^i$ and $\Delta_{\text{dyn},t}^i$
      Estimate critic confidence $g_{V,t}^i$ and compute fusion weight w_t^i
      Fuse the two corrections and obtain the agent-specific surrogate scale $k_{q,t}^i$
      Form the MAPPO objective with $\mathcal{L}_\sigma$, $\mathcal{L}_{\text{SGM}}$, and $\mathcal{L}_{\text{fr}}$
      Optimize $(\theta, \phi, \psi, \omega)$ with the modulated surrogate gradients
      Update historical return and value-loss statistics
    end for
  end for
end for
return (θ, ϕ)

A.2 Implementation and Experiment Details

A.2.1 Details of MARL Tasks

Experiments are conducted on three widely adopted cooperative MARL environments: the StarCraft Multi-Agent Challenge (SMAC) Samvelyan et al. (2019), Google Research Football (GRF) Kurach et al. (2020), and the Multi-Agent Particle Environment (MPE) Lowe et al. (2017), as illustrated in Fig. 4.



Figure 4: Environments used in our experiments: SMAC (left), GRF (middle), and MPE (right).

SMAC is a cooperative combat benchmark based on StarCraft II, where each unit is controlled by an independent agent. Each agent receives local observations containing information about nearby allied and enemy units, including relative positions, health states, and available actions. A global state with complete information about all units is provided to the centralized critic. Agents select from a discrete action space consisting of movement, attack, and stop actions. Rewards are calculated based on damage dealt to enemies, enemy elimination, and victory bonuses. Following the default SMAC setting, rewards are normalized such that the maximum episodic reward is approximately 20. The configurations of the selected SMAC scenarios are summarized in Table 2.

GRF is a multi-agent football simulation environment designed to evaluate cooperative decision-making among multiple players. Agents interact in football scenarios that require coordination

Table 2: Dimension of the observation space and the action space and the episodic length of SMAC

Task	Dimension of observation space	Dimension of action space	Episodic length
3m	48	9	60
8m	168	14	120
2s3z	120	11	120
1c3s5z	270	15	180
25m	950	31	150
3s5z	216	14	150
5m_vs_6m	98	12	70
8m_vs_9m	179	15	120
10m_vs_11m	243	17	150
2c_vs_64zg	342	70	400
3s_vs_5z	68	11	250
MMM2	322	18	180
3s5z_vs_3s6z	230	15	170
27m_vs_30m	1170	36	180
6h_vs_8z	140	14	150

through passing, shooting, movement, and defensive behaviors. Each player observes information related to positions, ball possession, movement directions, and surrounding game states. The reward combines scoring rewards with environment-defined shaping rewards. We adopt sparse reward settings, where positive rewards are mainly obtained from successful scoring events. Detailed configurations of the GRF scenarios are provided in Table 3.

Table 3: Dimension of the observation space and the action space and the episodic length of GRF

Task	Dimension of observation space	Dimension of action space	Episodic length
3 vs 1 (keeper)	26	19	150
Counterattack (easy)	30	19	150
Counterattack (hard)	34	19	150
Pass and shoot	22	19	150
Run, pass and shoot	30	19	150
Run to score	26	19	150
Run to score (keeper)	30	19	150

MPE is a lightweight multi-agent benchmark for studying coordination among agents in cooperative navigation tasks. Agents interact with other agents and landmarks, where successful collaboration requires reaching target locations while avoiding collisions. Each agent receives information about relative positions, velocities, and entity-related states. Rewards are calculated according to the distance between agents and targets, together with collision penalties. Table 4 summarizes the configurations of the evaluated MPE scenarios.

Table 4: Dimension of the observation space and the action space and the episodic length of MPE

Task	Dimension of observation space	Dimension of action space	Episodic length
Reference	11	5	25
Speaker-listener	11	5	25
Spread	18	5	25

A.2.2 INFRASTRUCTURE

All experiments are conducted on a computing platform equipped with an Intel Xeon Platinum 8352Y CPU, 2048 GB of RAM, and two NVIDIA GeForce RTX 4090 GPUs. Spike-BiST is implemented in Python using PyTorch, with SpikingJelly employed for the implementation and training of spiking neural networks.

A.2.3 EXPERIMENT DETAILS

All experiments are conducted based on the PyMARL Samvelyan et al. (2019) and MAPPO Yu et al. (2022) frameworks. GRSN Qin et al. (2025), EMU Na et al. (2024), and ICES Li et al. (2024a) are implemented in PyMARL, while Spike-BiST, SNN-MAPPO Yu et al. (2022), and ILC-SAN Chen et al. (2025) are implemented in MAPPO. All compared methods use the same environment versions, interaction budgets, number of evaluation episodes, and random-seed protocol within each benchmark. Hyperparameters follow the original papers and released implementations when available. The main settings of Spike-BiST are listed in Tables 5 and 6.

Table 5: Main hyperparameters of Spike-BiST.

Hyperparameter	Value	Hyperparameter	Value
Internal timesteps T_s	8	Leak factor β	0.95
Firing threshold V_{th}	1.0	Initial membrane potential	0
Reset mechanism	Soft reset	Surrogate function	Fast sigmoid
Population neurons P	8	Identity dimension d_{id}	1
EMA coefficient γ_{ema}	0.99	Center-shift coefficient η	0.8
Reference width $\sigma^0_{d,p}$	$1.2\delta_{\mu,d}$	Width regularization range $[\sigma_{\min}, \sigma_{\max}]$	$[10^{-6}, 1.0]$
Width regularization λ_σ	0.01	Firing-rate regularization λ_ρ	0.01
Firing-rate range $[\rho_{\min}, \rho_{\max}]$	$[0.05, 0.5]$	SGM regularization λ_{SGM}	0.01
Progress EMA coefficient	0.99	Return-variation EMA coefficient	0.99
Critic-loss EMA coefficient	0.99	Base-anchor coefficient λ_b	0.01
Initial surrogate scale k_0	1.0	Team-scale coefficient κ_R	5.0
Reward-change weight η_R	1.0	Surrogate-scale update	Every PPO mini-batch
Base-scale bounds $[k_0^{\min}, k_0^{\max}]$	$[1, 20]$	Final-scale bounds $[k^{\min}, k^{\max}]$	$[1, 20]$
Advantage amplitude a_0	0.3	Dynamics amplitude a_1	0.3
Advantage sensitivity η_A	0.5	Base attenuation λ_A	0.02
Critic-confidence sensitivity β_V	1.0	Confidence lower bound $g_{\min}$	0.05
Hypernetwork hidden dimension	64	Dynamics-network hidden dimension	32
Gate-network hidden dimension	32	MLP activation	ReLU

Table 6: Main hyperparameters of the MAPPO framework.

Hyperparameter	Value	Hyperparameter	Value
Actor learning rate	3×10^{-4}	Critic learning rate	5×10^{-4}
PPO clip ratio	0.2	PPO epochs	10
Discount factor γ	0.99	GAE parameter λ	0.95
Entropy coefficient c_H	0.01	Value-loss coefficient c_V	1.0
Rollout length	400	Mini-batches	2
Gradient clipping	5.0	Optimizer	Adam
Clipped value loss	Enabled	Huber threshold	10

Since spiking MARL methods remain limited, GRSN Qin et al. (2025) is included as a representative spiking MARL baseline for comparison with existing spiking multi-agent approaches. EMU Na et al. (2024) and ICES Li et al. (2024a) are selected as representative MARL baselines that enhance cooperative learning through additional exploration and individual learning mechanisms. To evaluate the impact of SNN architectures and surrogate gradient optimization, SNN-MAPPO Yu et al. (2022) is constructed by replacing the policy network in MAPPO with an SNN architecture. ILC-SAN Chen et al. (2025), proposed for single-agent spiking reinforcement learning, is extended to the

MAPPO framework for multi-agent evaluation. These baselines provide comparisons from different perspectives, including spiking MARL, conventional MARL policies, and SRL methods.

For comprehensive evaluation, experiments are conducted on a variety of maps with different coordination requirements determined by map difficulty. The win rate is evaluated over 32 test episodes for each training seed, with 5 independent random seeds. The learning curves are sampled at 100 evaluation points, and the shaded regions represent the 95% confidence intervals across different seeds.

For SMAC experiments, the version of $starcraft.py$ provided in MAPPO Yu et al. (2022) is adopted. All SMAC experiments are conducted on difficulty level 7 using StarCraft II version B69232 (SC2.4.6-Publish) in a Linux environment. The evaluated scenarios include 4 Easy tasks, 7 hard tasks, and 4 super hard tasks, covering different coordination difficulties and unit compositions. Easy and hard tasks are trained for 10 million environment steps, while super hard tasks are trained for 10 or 20 million environment steps according to the task setting. The GRF benchmark Kurach et al. (2020) is evaluated using gfootball version 2.10.1. Seven scenarios with sparse reward settings are selected, and each scenario is trained for 10 million environment steps. For the MPE benchmark Lowe et al. (2017), the environment is implemented based on PettingZoo with Gym-compatible interfaces. Three cooperative scenarios, including Reference, Speaker-listener, and Spread, are evaluated. The three scenarios are trained for 3, 2, and 5 million environment steps, respectively.

A.3 Further Experiment Results

A.3.1 Results on Additional Benchmarks

To further evaluate the performance of Spike-BiST across different task scenarios, additional experiments are conducted on SMAC, GRF, and MPE benchmarks. Spike-BiST is compared with the aforementioned methods under various tasks, and the results are summarized in Tables 7-9.

Table 7: Test win rate (%) on SMAC tasks.

Task	Difficulty	Spike-BiST	GRSN	EMU	ICES	SNN-MAPPO	ILC-SAN
3m	Easy	**100.0 ± 0.0**	99.4 ± 1.4	**100.0 ± 0.0**	99.0 ± 1.5	99.4 ± 1.4	99.4 ± 1.4
8m	Easy	**100.0 ± 0.0**	98.1 ± 2.8	**100.0 ± 0.0**	99.0 ± 1.5	**100.0 ± 0.0**	96.9 ± 2.6
2s3z	Easy	**100.0 ± 0.0**	95.6 ± 3.6	98.4 ± 1.8	**100.0 ± 0.0**	**100.0 ± 0.0**	98.1 ± 1.7
1c3s5z	Easy	**98.5 ± 2.0**	89.8 ± 5.9	98.1 ± 2.8	98.1 ± 3.2	98.1 ± 1.7	83.1 ± 1.8
25m	Hard	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	25.2 ± 4.8
3s5z	Hard	**98.1 ± 3.8**	85.6 ± 7.6	85.2 ± 8.2	94.9 ± 6.2	82.4 ± 9.1	73.1 ± 15.7
5m_vs_6m	Hard	85.9 ± 4.0	45.6 ± 11.2	**90.2 ± 5.6**	41.3 ± 7.2	77.5 ± 11.1	21.9 ± 6.0
8m_vs_9m	Hard	**98.1 ± 1.7**	65.8 ± 8.3	46.9 ± 9.1	18.9 ± 5.4	95.3 ± 6.2	11.6 ± 2.6
10m_vs_11m	Hard	**100.0 ± 0.0**	73.3 ± 11.6	75.0 ± 11.2	10.2 ± 9.3	83.4 ± 7.4	13.7 ± 5.6
2c_vs_64zg	Hard	93.0 ± 5.9	87.6 ± 8.8	83.4 ± 9.6	**96.9 ± 2.6**	74.4 ± 12.9	3.6 ± 1.4
3s_vs_5z	Hard	**100.0 ± 0.0**	53.1 ± 5.6	97.0 ± 4.0	73.4 ± 6.5	62.5 ± 5.7	47.3 ± 5.2
MMM2	Super Hard	90.0 ± 4.2	48.8 ± 17.9	**97.2 ± 3.1**	73.2 ± 12.2	68.2 ± 12.9	2.9 ± 6.4
3s5z_vs_3s6z	Super Hard	81.3 ± 12.1	63.8 ± 18.2	**85.6 ± 4.5**	83.6 ± 14.3	74.4 ± 13.1	7.8 ± 6.2
27m_vs_30m	Super Hard	**99.6 ± 1.2**	88.5 ± 5.8	78.0 ± 6.4	93.3 ± 6.4	83.2 ± 8.2	9.5 ± 6.4
6h_vs_8z	Super Hard	**69.3 ± 7.8**	62.5 ± 8.0	35.2 ± 9.8	57.8 ± 9.2	46.2 ± 10.2	3.1 ± 2.2

The test win rates on 15 SMAC tasks are reported in Table 7. Spike-BiST achieves the highest or tied-highest win rate on 11 tasks and the second-highest on three tasks, which demonstrates competitive performance across diverse scenarios. In particular, Spike-BiST reaches a 100.0% win rate on several easy and hard tasks, indicating its capability in solving diverse cooperative tasks. On the super hard MMM2 scenario, Spike-BiST achieves a win rate of 90.0%, which remains competitive but is lower than some ANN-based MARL methods. This gap may be related to the limited representation capacity of binary spike activations in complex coordination scenarios, where richer state representations and more flexible policy optimization are beneficial.

Table 8: Test goal rate (%) on GRF tasks.

Task	Spike-BiST	GRSN	EMU	ICES	SNN-MAPPO	ILC-SAN
3 vs 1 (keeper)	**72.7 ± 11.5**	65.3 ± 15.7	64.2 ± 15.2	48.3 ± 8.5	62.3 ± 11.3	28.6 ± 10.3
Counterattack (easy)	**64.0 ± 16.5**	37.5 ± 16.1	51.5 ± 19.4	31.3 ± 16.6	52.3 ± 18.9	16.8 ± 7.4
Counterattack (hard)	**49.8 ± 10.7**	18.7 ± 11.5	34.3 ± 13.0	47.2 ± 9.8	15.5 ± 10.1	0.8 ± 0.6
Pass and shoot	**93.5 ± 2.7**	80.9 ± 8.7	88.7 ± 3.1	68.8 ± 4.5	88.8 ± 3.3	55.4 ± 7.6
Run, pass and shoot	**69.1 ± 2.5**	52.0 ± 6.1	62.5 ± 3.1	47.2 ± 4.4	56.6 ± 18.6	39.5 ± 9.8
Run to score	92.2 ± 6.3	87.2 ± 9.7	87.5 ± 8.2	**93.3 ± 3.8**	87.3 ± 5.6	81.4 ± 7.2
Run to score (keeper)	**96.2 ± 4.4**	75.5 ± 9.6	93.2 ± 4.8	92.4 ± 5.6	81.6 ± 8.9	67.8 ± 8.2

Spike-BiST achieves competitive performance on the GRF benchmark, as shown in Table 8. Spike-BiST obtains the highest goal rate on most scenarios and achieves notable improvements on challenging tasks such as Counterattack hard and Pass and shoot. On the Run to score task, ICES achieves a slightly higher goal rate than Spike-BiST, indicating that different coordination strategies may benefit from specific task characteristics. The results demonstrate the effectiveness of Spike-BiST in handling complex cooperative decision-making tasks with state changes.

Table 9: Return score on MPE tasks.

Subtask	Spike-BiST	GRSN	EMU	ICES	SNN-MAPPO	ILC-SAN
Reference	-21.2 ± 1.2	**-18.4 ± 2.6**	-37.1 ± 7.6	-20.3 ± 6.3	-23.0 ± 1.3	-19.7 ± 2.2
Speaker-listener	**-13.0 ± 6.8**	-13.3 ± 1.2	-39.5 ± 11.1	-19.2 ± 3.3	-30.2 ± 1.7	-13.2± 4.2
Spread	**-112.8 ± 18.1**	-178.7 ± 17.4	-183.5 ± 13.2	-118.4 ± 5.8	-121.9 ± 11.2	-156.0 ± 23.6

Table 9 summarizes the return scores on MPE tasks. Spike-BiST achieves the highest returns on Speaker-listener and Spread, achieving better returns than most compared methods. On the Reference task, GRSN obtains a higher return than Spike-BiST, suggesting that different task settings may lead to different performance preferences among methods. Overall, the results across SMAC, GRF, and MPE demonstrate the effectiveness of Spike-BiST across diverse cooperative multi-agent reinforcement learning scenarios.

A.3.2 Further Ablation Studies

To isolate the effect of dual-channel adaptive population encoding from multi-agent interactions, we further evaluate the encoder on the MuJoCo Hopper and Walker2d tasks, as shown in Fig. 5(a) and Fig. 5(b). We compare dual-channel adaptive population encoding with adaptive and fixed encoding under the same policy architecture, training procedure, and hyperparameters. Each method is trained for one million environment interaction steps.

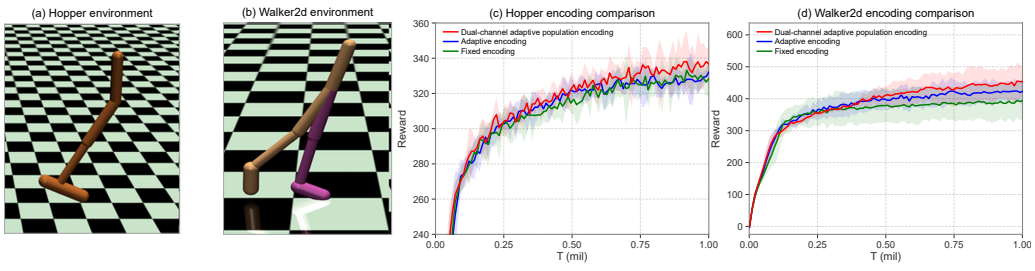

Figure 5: MuJoCo environments and encoding comparisons: (a) Hopper environment; (b) Walker2d environment; (c) Encoding comparison on Hopper; and (d) Encoding comparison on Walker2d.

As shown in Fig. 5(c), the three variants show similar learning trends during the early stage on Hopper. The dual-channel adaptive encoder achieves higher training rewards than fixed encoding during the later stage and shows an advantage over adaptive encoding. These results indicate that adaptive population representations can reduce the mismatch between fixed encoding parameters and changing observation distributions during policy training.

A larger difference is observed on Walker2d in Fig. 5(d). Fixed encoding reaches a reward plateau at an earlier stage, whereas the adaptive variants continue to gain training reward. The dual-channel adaptive encoder achieves the highest training reward during the middle and later stages. These results support the role of adaptive encoding in handling changes in observation distributions and show the benefit of the dual-channel design for policy optimization.