# Proxy Target: Bridging the Gap Between Discrete Spiking Neural Networks and Continuous Control

#### Zijie Xu

Institute for Artificial Intelligence Peking University Beijing, China 100871 xuzj32@gmail.com

#### **Zecheng Hao**

School of Computer Science Peking University Beijing, China 100871 haozecheng@pku.edu.cn

#### Tong Bu

Institute for Artificial Intelligence Peking University Beijing, China 100871 putong30@pku.edu.cn

# Jianhao Ding

School of Computer Science Peking University Beijing, China 100871 djh01998@stu.pku.edu.cn

#### Zhaofei Yu

Institute for Artificial Intelligence School of Computer Science Peking University Beijing, China 100871 yuzf12@pku.edu.cn

#### Abstract

Spiking Neural Networks (SNNs) offer low-latency and energy-efficient decision making through neuromorphic hardware, making them compelling for Reinforcement Learning (RL) in resource-constrained edge devices. Recent studies in this field directly replace Artificial Neural Networks (ANNs) by SNNs in existing RL frameworks, overlooking whether the RL algorithm is suitable for SNNs. However, most RL algorithms in continuous control are designed tailored to ANNs—including the target network soft updates mechanism—which conflict with the discrete, non-differentiable dynamics of SNN spikes. We identify that this mismatch destabilizes SNN training in continuous control tasks. To bridge this gap between discrete SNN and continuous control, we propose a novel proxy target framework. The continuous and differentiable dynamics of the proxy target enable smooth updates, bypassing the incompatibility of SNN spikes, stabilizing the RL algorithms. Since the proxy network operates only during training, the SNN retains its energy efficiency during deployment without inference overhead. Extensive experiments on continuous control benchmarks demonstrate that compared to vanilla SNNs, the proxy target framework enables SNNs to achieve up to 32% higher performance across different spiking neurons. Notably, we are the first to surpass ANN performance in continuous control with simple Leaky-Integrate-and-Fire (LIF) neurons. This work motivates a new class of SNN-friendly RL algorithms tailored to SNN's characteristics, paving the way for neuromorphic agents that combine high performance with low power consumption.

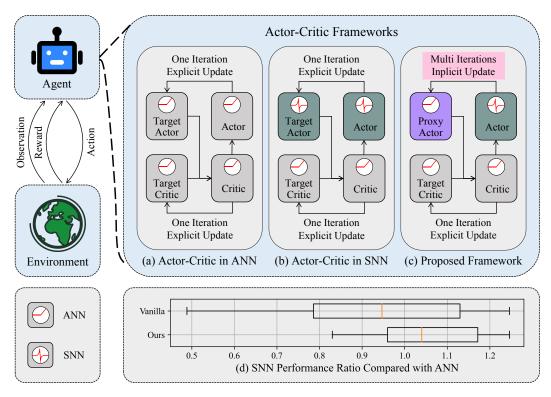


Figure 1: The oveall training framework and performance comparison. (a)-(c) are different training paradigms. (a): The Actor-Critic framework in ANN, (b): The Actor-Critic framework with spiking actor network, (c): The proposed framework with proxy target network for SNN. (d): Performance ratio of SNNs relative to ANNs across 5 random seeds and 5 different environments. The middle orange line denotes the median, the box spans from the first to the third quartile, and the whiskers extend to the farthest data within 1.5 inter-quartile range from the box.

#### 1 Introduction

With the combination of Artificial Neural Networks (ANNs), Reinforcement Learning (RL) has emerged as a cornerstone of modern artificial intelligence, achieving excellent results in various fields, including game playing [Mnih, 2013, Silver et al., 2016, Mnih et al., 2015], autonomous driving [Kiran et al., 2021, Sallab et al., 2017, Shalev-Shwartz et al., 2016] and large language model training [Ouyang et al., 2022, Bai et al., 2022, Shao et al., 2024]. Among these, continuous control problems have attracted significant research attention because of their close alignment with real-world scenarios and their strong connection to embodied AI and robotic applications [Kober et al., 2013, Gu et al., 2017, Brunke et al., 2022]. However, strict power budgets and computational constraints in some edge devices, such as drones, wearables, and IoT sensors, limit the deployment of ANN-based RL algorithms [Abadía et al., 2021, Tang et al., 2020, Yamazaki et al., 2022].

Inspired by biological neural systems, Spiking Neural Networks (SNNs) offer sparse and event-driven computation for ultra-low latency and energy consumption on neuromorphic hardwares [DeBole et al., 2019, Davies et al., 2018]. This efficiency is especially vital for the deployment on resource-constrained edge devices [Yamazaki et al., 2022]. Recently, there has been increasing attention to integrate SNNs with continuous control RL algorithms through the hybrid framework [Tang et al., 2020, 2021, Zhang et al., 2024, Chen et al., 2024a, Zhang et al., 2022, Chen et al., 2024b], where the spiking actor network (SAN) is co-trained with the ANN critic network by Spatial-temporal Backpropagation [Wu et al., 2018], as shown in Fig.1(b). With specific hyper-parameters settings, SAN has the potential to outperform ANNs in solving challenging continuous control problems.

However, in these works, SNNs are simply retrofitted into existing ANN-centric RL frameworks, without modifications to the RL algorithms. Since ANN and SNN exhibit different dynamics and characteristics, it is doubtful whether the RL algorithms designed for ANNs are suitable for SNNs. Are there any mismatches between the discrete dynamics of SNNs and the continuous control RL algorithms?

2

Specifically, most continuous control RL algorithms rely heavily on the target network soft update mechanism to stabilize training by reducing sudden changes in the optimization goal. This process relies on a continuous output drift in the target network, which inherently conflicts with the non-differentiable, binary nature of SNN. This mismatch can cause abrupt shifts in the optimization objective, potentially hindering the convergence of the learning process. As a result, the performance of existing SNN-based RL models becomes highly sensitive to random seed initialization, often leading to divergent outcomes. This instability also undermines the reliability of the model, making it challenging to deploy in real-world applications.

To address the mismatching issue between continuous control and discrete spikes, we propose a proxy target framework for SNNs based on the Actor-Critic framework (Fig.1(c)). We replace the target actor network with a novel proxy actor network. Compared to the vanilla SNN target network, the proxy network has continuous dynamics and is updated implicitly. The proxy target network can alter its output smoothly and continuously, stabilizing the learning process and improving the performance, as demonstrated in Fig.1(d). Since the proxy target network is only used for auxiliary training, the proposed approach retains SNN's advantages of low-latency and energy efficiency during inference in real world applications. Our main contributions are summarized as follows:

- We discover the mismatch between the discrete SNNs and the continuous target network soft update mechanism. Due to the binary and non-differentiable nature of SNN outputs, the target SNN may exhibit abrupt, non-smooth changes during training. These discontinuities introduce instability in the optimization process, making the training highly sensitive to random seed initialization and degrading performance.
- We propose the proxy target framework to address the mismatching issue. The proxy target replaces the target network by using a continuous activation function, producing smooth and continuous output drift that stabilizes training.
- To update the continuous proxy network towards the discrete SNN, we propose a novel gradient-based mechanism derived from the goal of approaching the output of the online SNN. This mechanism addresses approximation errors between the proxy target and online SNN, giving precise optimization goals for training the RL algorithms.
- Extensive experiments demonstrate the proxy network's ability to enhance performance of different spiking neurons by 5-32%. To the best of our knowledge, we are the first to surpass ANN's performance in continuous control with a simple LIF neuron.

# 2 Related works

# 2.1 Learning rules of SNN-based RL

**Synaptic plasticity.** Inspired by the plasticity of synapses in biological neural system, some works integrates SNNs into RL by developing reward-modulated spike-timing-dependent plasticity to simulate brain intelligence [Florian, 2007, Frémaux and Gerstner, 2016, Gerstner et al., 2018, Frémaux et al., 2013, Yang et al., 2024]. These approaches are biologically plausible and energy-efficient, but have limited performance in complicated tasks.

**ANN-SNN conversion.** With recent advances in ANN-based DRL, Patel et al. [2019], Tan et al. [2021], Kumar et al. [2025] converted a well-trained Deep-Q-Network [Mnih, 2013, Mnih et al., 2015] to SNN. These conversion-based approaches improved both robustness and energy efficiency in decision making, but require ANN pre-taining.

**Gradient based direct training.** To directly train SNNs in RL without ANN pre-taining, Liu et al. [2022], Chen et al. [2022], Qin et al. [2022] use Spatio-Temporal Backpropagation (STBP) [Wu et al., 2018] to train a Deep-Q-Network [Mnih, 2013, Mnih et al., 2015], while Bellec et al. [2020] proposed e-prob with eligibility trace that learns a policy function by the Policy Gradient algorithm [Sutton et al., 1999]. These direct-training approaches achieve competitive scores in discrete action spaces, but cannot be implemented in continuous control tasks.

## 2.2 Hybrid framework of spiking actor network.

In continuous control tasks where the action space is continuous, the hybrid framework are widely studied. Tang et al. [2020] first proposed an SNN as a spiking actor network, which is co-trained

with an ANN as a deep critic network in the Actor-Critic framework [Konda and Tsitsiklis, 1999]. Then, Tang et al. [2021] found that population encoding achieves the best performance for the spiking actor network, . After that, various improvements have been proposed to enhance performance, such as upgrading neuron dynamics [Zhang et al., 2022], incorporating lateral connections [Chen et al., 2024a], utilizing bio-plausible topologies [Zhang et al., 2024], integrating dynamic thresholds [Ding et al., 2022], combining with noisy parameters [Chen et al., 2024b].

These approaches are reported to exceed ANNs performance with the same RL algorithms, however, there are two drawbacks. First, the SNN's neurons and architectures are complex, i.e. current-based leaky integrate and fire neuron or even more complex 2nd order neurons. Second, the integrated RL algorithms are not modified for SNN's dynamics.

# 3 Preliminary

#### 3.1 Reinforcement Learning

Reinforcement Learning (RL) involves an agent interacting with the environment. At every time step, the agent observes the current state s and performs an action a, the environment gives a reward r and transfers to the next state s'. The agent's goal is to learn an optimal policy  $\pi_{\phi}$  with parameters  $\phi$  that maximizes the expected return  $J(\phi)$ .

In continuous control, the action space is a continuous vector indicating the extent of each action (e.g., torque). Most of the continuous control algorithms utilize the Actor-Critic framework with deterministic policy [Sutton and Barto, 2018], where the actor network  $\pi$  performs action  $a = \pi_{\phi}(s)$ , while the critic network rates this action and output a Q-value  $Q_{\theta}(s,a)$  with parameters  $\theta$  [Konda and Tsitsiklis, 1999].

The parameters in the actor network are updated by deterministic policy gradient algorithm [Silver et al., 2014]:

$$\nabla_{\phi} J(\phi) = \mathbb{E} \left[ \nabla_{a} Q_{\theta}(s, a) \mid_{a=\pi(s)} \nabla_{\phi} \pi_{\phi}(s) \right]. \tag{1}$$

The critic network is updated by temporal-difference learning (TD) [Sutton, 1988] according to the Bellman equation [Bellman, 1966]:

$$Q_{\theta}(s, a) \leftarrow y,$$

$$y = r + \gamma Q_{\theta'}(s', a'), a' = \pi_{\phi'}(s').$$
(2)

# 3.2 Target network soft update

Noticing that  $Q_{\theta'}$  and  $\pi_{\phi'}$  in Eq.(2) are the target critic network and the target actor network respectively. Target network has the same architecture as the network being trained online. The parameters of the target network are updated explicitly by the Polyak function with a target smoothing factor  $\tau$ :

$$\begin{aligned}
\phi' &\leftarrow \tau \phi + (1 - \tau) \phi', \\
\theta' &\leftarrow \tau \theta + (1 - \tau) \theta'.
\end{aligned} (3)$$

The target network soft update is so important that almost all continuous control algorithms use it. As shown in Eqs.(1,2) the actor network and the critic network are co-trained jointly with the sense of bootstrapping. This interdependence between the actor and the critic causes oscillations in both networks. The target network is designed to weaken the interaction between the online networks, alleviating bootstrapping and oscillations by remaining a relatively steady output.

# 3.3 Spiking Neural Network

**Spiking neuron model** In SNN, at every time step, each neuron integrates received presynaptic spikes into membrane potential and fires a spike if its membrane potential is higher than a threshold. The Leaky Integrate and Fire (LIF) neuron [Gerstner and Kistler, 2002] is one of the most commonly used spiking neurons. The dynamics of LIF neurons can be written as follows:

$$I_t^l = W^l S_t^{l-1} + b^l, (4)$$

$$H_t^l = \lambda V_{t-1}^l + I_t^l, \tag{5}$$

$$S_t^l = \Theta(H_t^l - V_{th}),\tag{6}$$

$$V_t^l = (1 - S_t^l) \cdot H_t^l + S_t^l \cdot V_{\text{reset}},\tag{7}$$

where I is the input current, H is the accumulated membrane potential, S is the binary output spike, V is the membrane potential after the firing process.  $V_{th}$ , W and b are the weights and the biases,  $V_{\text{reset}}$  and  $\lambda$  are the threshold voltage, the reset voltage and the membrane leakage parameter, respectively. All subscripts  $(\cdot)_t$  and all superscripts  $(\cdot)^l$  denote time step t and layer l respectively.  $\Theta(\cdot)$  is the Heaviside function.

**Spiking actor network.** The spiking actor network (SAN) consists of a population encoder with Gaussian reception fields [Tang et al., 2021], a multi layer SNN with population output and a decoder that takes the membrane potential of non-firing neurons as the output [Zhang et al., 2022]. The SAN is learned by spatio-temporal backpropagation (STBP) with a rectangular surrogate gradient function [Wu et al., 2018]. Detailed forward propagation and backpropagation are shown in the Appendix.

# 4 Methodology

In this section, we propose a novel proxy target framework. Section 4.2 first introduces the issue of discrete output in the SNN target network, then proposes the proxy target network with continuous dynamics to address this problem. To let the proxy network approach the online SNN, Section 4.2 proposes an implicit update mechanism and derives it into a gradient-based optimization process. Section 4.3 shows the overall training framework of the proposed methods.

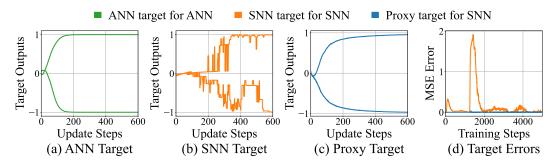


Figure 2: Effects of different update mechanisms. (a)-(c) are the output of different target network during the update, where different lines denote different output dimensions. Each line denote one dimension of the output vector normalized to (-1,1) by the tanh function. (d) shows the mean squared errors between the output of proxy target network (or SNN target network) and SNN actor during dynamic training in the InvertedDoublePendulum-v4 environment. The training procedure will be proposed in Section 4.3.

#### 4.1 Addressing discrete targets by proxy network

Performance degradation due to discrete target outputs. In the standard Actor-Critic framework, the target network is updated explicitly using the soft weight update, typically implemented via the Polyak averaging function (Eq.3), to ensure smooth transitions in output during training. This strategy relies on a critical assumption: if a network's parameters are updated gradually, its outputs will also change gradually with given fixed inputs. This assumption holds in ANNs due to the continuous nature of activation functions. However, the firing function of spiking neurons is non-differentiable and produces binary spike outputs. As a result, even small changes in network weights can cause abrupt and discontinuous shifts in the output. This violation of the smoothness assumption causes the SNN-based target network to produce discrete outputs during training, undermining the effectiveness of the soft update mechanism and destabilizing the learning process.

To further illustrate this point, we empirically show that soft weight updates are not well-suited for the target SNN architecture. We construct various target networks corresponding to their well-trained online network, ensuring that each target network shares the same neuron model and architecture as the online counterpart. Then the parameters in the target network are updated according to Eq.3 with  $\tau=0.005$ , while the parameters in the online network are frozen. Fig.2(a) and (b) present the output

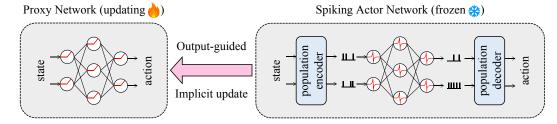


Figure 3: The network architecture of the proposed proxy network and the spiking actor network. The proxy network is updated implicitly by approaching the output of the spiking actor network.

trajectories of the target networks during soft updates for the ANN and SNN cases, respectively. After sufficient update steps, all target network outputs eventually converge to the online networks. However, the SNN target exhibits frequent discrete shifts (Fig.2(b)), resulting in more unstable and uneven output trajectories. These discontinuities violate the goal of the target network soft update, which relies on smooth and continuous output changes, as clearly observed in the ANN case in Fig.2(a). This further induces abrupt changes in the optimization target of the online critic network, leading to oscillations and unstable learning in the Actor-Critic algorithm [Fujimoto et al., 2018].

**Smoothing target outputs by proxy network.** One of the key requirements of the target network soft update is that the output of the target network must change smoothly and continuously. However, this requirement natively contradicts the discrete dynamics of SNNs. Thus, we propose a proxy network that replace the undifferentiable spiking neurons of target SNN by continuous activation function of ANN.

Since the proxy network exhibits continuous and differentiable dynamics, there will be no more discrete output shifts during update. To further demonstrate it, Fig.2(c) shows the output of the proxy target network during the update (the update process will be provided in Section 4.3), with the online network parameters frozen. The proxy target network updates its output smoothly and continuously, as similar to the target network for ANN in Fig.2.(a). This result demonstrates that the proposed proxy network is capable of addressing the issue of discrete target outputs and producing soft updates.

#### 4.2 Addressing appropriation errors by implicit updates

**Performance degradation due to appropriation errors.** In RL algorithms, the target network is expected to approximate the online network and eventually converge to the same output. However, simply replacing the spiking neuron with ReLU activations results in a target network with continuous dynamics that struggles to match the discrete behavior of the online SNN. As a result, the output discrepancy between the proxy network and the online SNN induces the non-negligible appropriation errors, leading to imprecise optimization goals that potentially degrades overall performance.

Aligning proxy network by implicit updates. As the approximation errors cannot be eliminated by explicitly copying the online SNN's weights, we propose an implicit target update method. Different from the previous explicit soft target update that directly update the model parameter, the proposed method implicitly calculate the update value so as to gradually reduce the output approximation error between the online SNN and proxy target, as shown in Fig.4.2. Supposing the proxy actor  $\pi_{\phi'}^{\text{Proxy}}$  has weights  $\phi'$  and the spiking actor network  $\pi_{\phi}^{\text{SNN}}$  has weights  $\phi$ , for each input state s, the output of the proxy network should be updated as:

$$\pi_{\phi'}^{\text{Proxy}}(s) \leftarrow (1 - \tau) \cdot \pi_{\phi'}^{\text{Proxy}}(s) + \tau \cdot \pi_{\phi}^{\text{SNN}}(s). \tag{8}$$

Since it is difficult to compute the exact values of  $\phi'$  after the update, we use gradient-based optimization to achieve similar effect with Eq.8:

$$\phi' \leftarrow \phi' + \tau \left( \pi_{\phi}^{\text{SNN}}(s) - \pi_{\phi'}^{\text{Proxy}}(s) \right) \nabla_{\phi'} \pi_{\phi'}^{\text{Proxy}}(s)$$
 (9)

$$= \phi' - \frac{\tau}{2} \nabla_{\phi'} \left\| \pi_{\phi'}^{\text{Proxy}}(s) \right\} - \pi_{\phi}^{\text{SNN}}(s) \right\|_{2}^{2}, \tag{10}$$

where  $\|\cdot\|_2^2$  denotes the squared L2 norm of the vector. Thus, the proxy network can be updated by gradient descent that minimizing the loss:

$$L_{proxy} = \frac{1}{N} \sum_{i=1}^{N} \left\| \pi_{\phi'}^{\text{Proxy}}(s_i) - \pi_{\phi}^{\text{SNN}}(s_i) \right\|_{2}^{2}, \tag{11}$$

where N denotes the batch size,  $s_i$  is the i-th input state sample from the replay buffer in RL algorithm. It is worth noting that the proxy actor is updated for K iterations during each proxy update episode due to the difficulty in approximating the discrete outputs of the online SNN.

Since the proxy is a multilayer feedforward network, which is an universal approximator [Hornik et al., 1989], it is able to reach the same output of the SNN actor by minimizing the loss in Eq.11. To further demonstrate it, Fig.2(d) shows the mean squared output gap between the SNN actor and the proxy network during dynamic training. The SNN target sometimes fails to follow the output of the online SNN, while the proxy network outputs similar to the SNN actor all the time. These results demonstrate that with the proposed updating mechanism, the proxy network is able to address approximation errors and give precise optimization goals for RL algorithms.

## 4.3 Overall training framework

The proposed proxy target framework is shown in Fig.1(c). The proxy actor network contains continuous activations that replace the discontinuous SNN target actor network. Instead of explicit parameters update, the proxy actor is updated implicitly by approaching the output of the SNN actor, which is realized by minimizing the loss in Eq.11. These modifications enable the proxy actor to smoothly and continuously alter its output towards the SNN actor, stabilizing the training process in the Actor-Critic framework.

Besides, the target critc is updated explicitly by copying weights using the Polyak function since the critic network and the critic target are both ANN. Detailed pseudo codes for general training framework is proposed in the Appendix.

Fig.2(c) and (d) demonstrate that the proxy actor network not only updates continuously, but also approaches the outputs of the SNN actor. Noticing that there are minor fluctuations in the proxy network output (Fig.2(c)), which are very similar to traditional soft target update for ANNs with a small amount of noise injection, a technique adopted in DRL to mitigate overfitting in value estimate [Fujimoto et al., 2018].

It is worth noting that the proposed mechanism remains the energy-efficiency of SNN, as the proxy network and the critic network are only used for assisting training, bringing no computational overhead in deployment.

# 5 Experiments

# 5.1 Experiments setup

The proposed proxy target framework (PT) was evaluated across different environments of various MuJoCo environments [Todorov et al., 2012, Todorov, 2014b] in the OpenAI Gymnasium benchmarks [Brockman, 2016, Towers et al., 2024], including InvertedDoublePendulum (IDP) [Todorov, 2014a], Ant [Schulman et al., 2015], HalfCheetah [Wawrzyński, 2009], Hopper [Erez et al., 2012] and Walker. All environment setups used the default configurations without modifications.

The experiments are carried out with different spiking neuron models, such as LIF, current-based LIF neuron (CLIF) proposed in Tang et al. [2021], and dynamic neuron (DN) proposed in Zhang et al. [2022]. The LIF and CLIF neuron parameters are as same as the work in Tang et al. [2021] (ignore the current leakage parameter in LIF), while the DN parameters are determined by the pre-learning process in Zhang et al. [2022].

We tested the proposed algorithm in conjunction with the TD3 algorithm [Fujimoto et al., 2018], all detailed parameter settings are shown in the Appendix. For a fair comparison, all spiking actor networks have the same architecture and use the same encoding and decoding scheme shown in the Appendix. All data in this section are our reproduced results.

## 5.2 Performance analysis

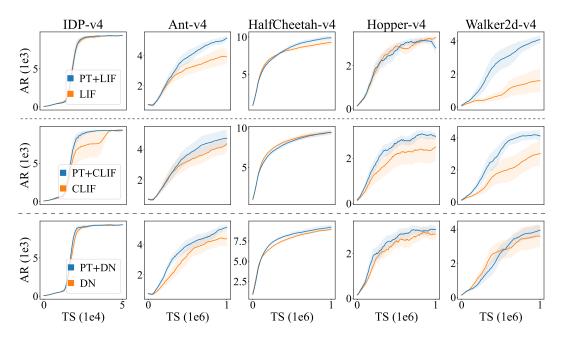


Figure 4: Learning curves of utilizing the proxy target framework in the LIF neuron, CLIF neuron and dynamic neuron (DN). The PT represents the proxy target framework, AR denotes average returns. and TS is training steps. The shaded region represents half a standard deviation over 5 different seeds. Curves are uniformly smoothed for visual clarity.

**Increasing performance for different spiking neurons.** Fig.4 shows the learning curves of the proposed proxy target framework and the vanilla Actor-Critic framework with different spiking neurons. The proxy target framework improves the performance of different spiking neurons, demonstrating its general applicability to deliver both faster convergence and higher final returns in different spiking neurons and different environments.

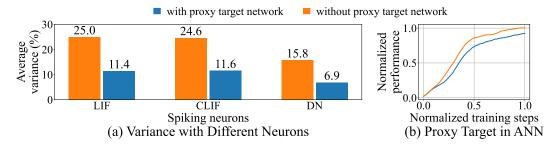


Figure 5: (a) Average variance of different neurons after training. The average variance is computed by averaging the standard deviation ratio with 5 seeds, across all environments. (b) Normalized learning curves across all environments of the ANN integrated with the proposed proxy network across all environments. The performance and training steps are normalized linearly to (0,1). Curves are uniformly smoothed for visual clarity.

**Improving stability for different spiking neurons.** Fig.5(a) shows the performance variance of the proxy target framework and the vanilla Actor-Critic framework with different spiking neurons. The proxy target framework reduces the variance of different spiking neurons, demonstrating its capability to stabilize training. This is crucial for real-world deployments, where retraining costs are high and consistent behavior is required.

Table 1: Max average returns over 5 random seeds with different spiking neurons, and the average proformance gain against ANN baseline, where  $\pm$  denotes one standard deviation.

Method	IDP-v4	Ant-v4	HalfCheetah-v4	Hopper-v4	Walker2d-v4	APG
ANN (TD3) Vanilla LIF Tang et al. [2021] Zhang et al. [2022] Chen et al. [2024a]	$7503 \pm 3713$ $9347 \pm 1$ $9351 \pm 1$ $9350 \pm 1$ $9352 \pm 1$	$4770 \pm 1014$ $4294 \pm 1170$ $4590 \pm 1006$ $4800 \pm 994$ $5584 \pm 272$	$\begin{array}{c} 10857 \pm 475 \\ 9404 \pm 625 \\ 9594 \pm 689 \\ 9147 \pm 231 \\ 9222 \pm 615 \end{array}$	$3410 \pm 164$ $3520 \pm 94$ $2772 \pm 1263$ $3446 \pm 131$ $3403 \pm 148$	$4340 \pm 383$ $1862 \pm 1450$ $3307 \pm 1514$ $3964 \pm 1353$ $4200 \pm 717$	0.00% $-10.54%$ $-6.66%$ $0.37%$ $4.64%$
PT-CLIF PT-DN PT-LIF	$9351 \pm 1$ $9350 \pm 1$ $9348 \pm 1$	$5014 \pm 1074$ $5400 \pm 277$ $5383 \pm 250$	$9663 \pm 426$ $9347 \pm 666$ $10103 \pm 607$	$3526 \pm 112$ $3507 \pm 144$ $3385 \pm 157$	$4564 \pm 555$ $4277 \pm 650$ $4314 \pm 423$	5.46% $5.06%$ $5.84%$

**Exceeding state-of-the-art.** To further illustrate the performance gain the average performance gain (APG) is defined as:

$$APG = \left(\frac{1}{|\text{envs}|} \sum_{\text{env} \in \text{envs}} \frac{\text{proformance(env)}}{\text{baseline(env)}} - 1\right) \cdot 100\%,\tag{12}$$

where |envs| denotes the total numbers of environments, proformance(env) and baseline(env) are the performance of the algorithm and the baseline in that particular environment. Tab.1 compares our proxy target framework with ANN-based RL and other state-of-the-art SNN-based RL algorithms, including pop-SAN [Tang et al., 2021], MDC-SAN [Zhang et al., 2022] and ILC-SAN [Chen et al., 2024a]. With the proxy network, a simple LIF-based SNN not only outperforms all other algorithms, including well-designed SNNs based on complex neuron dynamics [Zhang et al., 2022] and connection architectures [Chen et al., 2024a], but also surpasses the performance of a standard ANN. These rusults demonstrates the efficiency of the proxy target framework.

**Simple neuron makes the best.** It is interesting to find out that in the proxy target framework, the simplest LIF neuron performs the best. This contrasts with other wide-seen situations in which complex neurons perform better. That is because it is more difficult for the proxy network to learn the reflections of the SNN with more complex neuron dynamics.

**SNN-friendly design.** Fig.5(b) shows the normalized performance of ANN with and without the proxy network. The proxy target framework cannot improve the performance in ANNs, indicating that the performance gain in SNNs is due to the SNN-friendly design, rather than a better RL algorithm.

Table 2: Energy consumptions of different tasks per inference for the spiking actor network with LIF neurons, where the energy unit is nano-joule (nJ).

Method	IDP-v4	Ant-v4	HalfCheetah-v4	Hopper-v4	Walker2d-v4	Average
ANN (TD3)	850.85	931.20	892.80	864.00	892.80	886.33
Vanilla LIF	8.14	11.78	15.13	7.21	18.82	12.21
PT-LIF	9.01	12.18	13.46	6.86	13.93	<b>11.09</b>

Energy consumptions. We test the energy consumption with our proxy target framework, as shown in Tab.2. The comparison includes a traditional ANN-based TD3 model, a baseline spiking actor network using vanilla LIF neurons, and our proposed proxy-target LIF (PT-LIF) model. The energy is calculated as the same way of Merolla et al. [2014], where floating-point operation (FLOP) costs 12.5pJ and synaptic operation (SOP) costs 77fJ [Qiao et al., 2015, Hu et al., 2021]. As shown, the ANN (TD3) model consumes significantly more energy, while both spiking models demonstrate dramatically lower energy consumption, by more than two orders of magnitude. Specifically, our proposed PT-LIF model achieves the lowest average consumption while achieving better stability and performance. These results highlight the superior energy efficiency of the proposed method, making it a compelling candidate for deployment on energy-constrained platform.

# 6 Conclusion

In this work, we identified a critical mismatch between the discrete dynamics of SNNs and the continuous requirement of the target network soft update mechanism in the Actor-Critic framework.

To address this, we proposed a novel proxy target framework that enables smooth target updates and faster convergence. Experimental results demonstrate that the proxy network can stabilize training and improve performance, enabling simple LIF neurons to surpass ANN performance in continuous control.

In contrast to previous works which retrofit SNNs into ANN-centric RL frameworks, this work opens a door to investigate and design SNN-friendly RL algorithms which is tailored for SNN's specific dynamics. In the future, more SNN-specific adjustments could be applied to SNN-based RL algorithms to improve performance and energy-efficient in the real-world, resource-constrained RL applications.

**Limitation.** While this work designs a proxy target framework that is suitable for SNN-based RL, it still remains at the simulation level. The next step may involve implementing it on edge devices and making decisions in the real world.

#### References

- Ignacio Abadía, Francisco Naveros, Eduardo Ros, Richard R Carrillo, and Niceto R Luque. A cerebellar-based solution to the nondeterministic time delay problem in robotic control. *Science Robotics*, 6(58):eabf2756, 2021.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.
- Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature communications*, 11(1):3625, 2020.
- Richard Bellman. Dynamic programming. science, 153(3731):34–37, 1966.
- G Brockman. Openai gym. arXiv preprint arXiv:1606.01540, 2016.
- Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):411–444, 2022.
- 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, 2024a.
- Ding Chen, Peixi Peng, Tiejun Huang, and Yonghong Tian. Noisy spiking actor network for exploration. *arXiv preprint arXiv:2403.04162*, 2024b.
- Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 2018.
- Michael V DeBole, Brian Taba, Arnon Amir, Filipp Akopyan, Alexander Andreopoulos, William P Risk, Jeff Kusnitz, Carlos Ortega Otero, Tapan K Nayak, Rathinakumar Appuswamy, et al. TrueNorth: Accelerating from zero to 64 million neurons in 10 years. *Computer*, 2019.
- Jianchuan Ding, Bo Dong, Felix Heide, Yufei Ding, Yunduo Zhou, Baocai Yin, and Xin Yang. Biologically inspired dynamic thresholds for spiking neural networks. *Advances in neural information processing systems*, 35:6090–6103, 2022.
- Tom Erez, Yuval Tassa, and Emanuel Todorov. Infinite-horizon model predictive control for periodic tasks with contacts. *Robotics: Science and Systems VII*, 2012.
- Răzvan V Florian. Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. *Neural computation*, 19(6):1468–1502, 2007.

- Nicolas Frémaux and Wulfram Gerstner. Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Frontiers in neural circuits*, 9:85, 2016.
- Nicolas Frémaux, Henning Sprekeler, and Wulfram Gerstner. Reinforcement learning using a continuous time actor-critic framework with spiking neurons. *PLoS computational biology*, 9(4): e1003024, 2013.
- Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actorcritic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.
- Wulfram Gerstner and Werner M Kistler. Spiking neuron models: Single neurons, populations, plasticity. Cambridge university press, 2002.
- Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules. *Frontiers in neural circuits*, 12:53, 2018.
- Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In 2017 IEEE international conference on robotics and automation (ICRA), pages 3389–3396. IEEE, 2017.
- Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- Yangfan Hu, Huajin Tang, and Gang Pan. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):5200–5205, 2021.
- B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE transactions on intelligent transportation systems*, 23(6):4909–4926, 2021.
- Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.
- Vijay Konda and John Tsitsiklis. Actor-critic algorithms. Advances in neural information processing systems, 12, 1999.
- Aakash Kumar, Lei Zhang, Hazrat Bilal, Shifeng Wang, Ali Muhammad Shaikh, Lu Bo, Avinash Rohra, and Alisha Khalid. Dsqn: Robust path planning of mobile robot based on deep spiking q-network. *Neurocomputing*, 634:129916, 2025.
- 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, 2022.
- Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spikingneuron integrated circuit with a scalable communication network and interface. *Science*, 345 (6197):668–673, 2014.
- Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

- Devdhar Patel, Hananel Hazan, Daniel J Saunders, Hava T Siegelmann, and Robert Kozma. Improved robustness of reinforcement learning policies upon conversion to spiking neuronal network platforms applied to atari breakout game. *Neural Networks*, 120:108–115, 2019.
- Ning Qiao, Hesham Mostafa, Federico Corradi, Marc Osswald, Fabio Stefanini, Dora Sumislawska, and Giacomo Indiveri. A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses. Frontiers in Neuroscience, 9:141, 2015.
- Lang Qin, Rui Yan, and Huajin Tang. A low latency adaptive coding spiking framework for deep reinforcement learning. *arXiv* preprint arXiv:2211.11760, 2022.
- Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *arXiv* preprint arXiv:1704.02532, 2017.
- John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438, 2015.
- Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr, 2014.
- David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.
- Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3: 9–44, 1988.
- Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. MIT press, 2018.
- Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.
- Weihao Tan, Devdhar Patel, and Robert Kozma. Strategy and benchmark for converting deep q-networks to event-driven spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 9816–9824, 2021.
- Guangzhi Tang, Neelesh Kumar, and Konstantinos P Michmizos. Reinforcement co-learning of deep and spiking neural networks for energy-efficient mapless navigation with neuromorphic hardware. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6090–6097. IEEE, 2020.
- Guangzhi Tang, Neelesh Kumar, Raymond Yoo, and Konstantinos Michmizos. Deep reinforcement learning with population-coded spiking neural network for continuous control. In *Conference on Robot Learning*, pages 2016–2029. PMLR, 2021.
- Emanuel Todorov. Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in mujoco. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 6054–6061. IEEE, 2014a.
- Emanuel Todorov. Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in mujoco. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 6054–6061. IEEE, 2014b.

- Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In 2012 IEEE/RSJ international conference on intelligent robots and systems, pages 5026–5033. IEEE, 2012.
- Mark Towers, Ariel Kwiatkowski, Jordan K Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, KG Arjun, et al. Gymnasium: A standard interface for reinforcement learning environments. *CoRR*, 2024.
- Paweł Wawrzyński. A cat-like robot real-time learning to run. In *Adaptive and Natural Computing Algorithms: 9th International Conference, ICANNGA 2009, Kuopio, Finland, April 23-25, 2009, Revised Selected Papers 9*, pages 380–390. Springer, 2009.
- Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.
- Kashu Yamazaki, Viet-Khoa Vo-Ho, Darshan Bulsara, and Ngan Le. Spiking neural networks and their applications: A review. *Brain sciences*, 12(7):863, 2022.
- 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*, 2024.
- Duzhen Zhang, Tielin Zhang, Shuncheng Jia, and Bo Xu. Multi-sacle dynamic coding improved spiking actor network for reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 59–67, 2022.
- Duzhen Zhang, Qingyu Wang, Tielin Zhang, and Bo Xu. Biologically-plausible topology improved spiking actor network for efficient deep reinforcement learning. *arXiv preprint arXiv:2403.20163*, 2024.