LEARNABLE MEMBRANE TIME CONSTANTS FOR STRUCTURE DYNAMICS IN RECURRENT SPIKING NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

In recurrent spiking neural networks (RSNNs), adapting connectivity structure represents an important means of RSNN optimization. However, their computational capabilities depend not only on connectivity structure but also on intrinsic neuronal dynamics. Previous studies have shown that heterogeneous neuronal timescales can improve learning in SNNs. Although existing approaches have made neuronal dynamics learnable, such adaptation is typically performed within a fixed connectivity structure. This raises the challenge of jointly adapting connectivity structure and heterogeneous neuronal dynamics without relying on backpropagation. To this end, we propose Directional Credit guided Forward Optimization (DCFO), a biologically inspired, backpropagation-free learning framework that jointly adapts connectivity structure and neuron-specific membrane time constants. DCFO constructs candidate directions separately in the connectivity parameter space and the neuron-specific membrane-time-constant space, providing directional credit for each. The resulting directional credit guides forward-gradient updates in the two parameter spaces, enabling structural–dynamical co-adaptation. We evaluate DCFO on three Brax continuous-control tasks, including the 17-DoF Humanoid. The proposed method outperforms the representative Evolving Connectivity baseline across a range of tasks and achieves performance competitive with deep recurrent neural network baselines. The learned neuronal dynamics further exhibit task-dependent heterogeneous neuronal timescales. These results demonstrate that DCFO enables effective structural–dynamical co-adaptation in RSNNs without relying on backpropagation.

1 INTRODUCTION

Recurrent spiking neural networks (RSNNs) are biologically inspired computational models that process temporal information through recurrent connectivity and spiking dynamics(Bellec et al., 2018b; 2020). Recurrent connectivity organizes information flow across the network and thereby shapes RSNN computation. Accordingly, adapting connectivity structure represents an important means of RSNN optimization(Chen et al., 2021; 2022; Wang et al., 2023). The computational capabilities of neural systems depend not only on connectivity structure but also on intrinsic neuronal dynamics. Connectivity determines the pathways through which information propagates, whereas heterogeneous neuronal timescales can improve learning in SNNs(Perez-Nieves et al., 2021). A more complete form of task adaptation should therefore allow connectivity structure and heterogeneous neuronal dynamics to co-adapt according to task demands.

Existing approaches have made neuronal dynamics learnable together with synaptic weights(Fang et al., 2021; Yin et al., 2021; Yao et al., 2022), but such adaptation is typically performed within a fixed connectivity structure. In biologically inspired structural optimization, however, the connectivity structure itself is adaptive and changes according to task demands. As the connectivity structure changes, neuronal dynamics must be optimized within a continuously changing structural context. Jointly adapting discrete connectivity structure and continuous neuronal dynamics therefore poses a distinct challenge for learning without relying on surrogate-gradient-based backpropagation(Neftci et al., 2019). This raises a key question: can RSNNs jointly adapt their connectivity structure and heterogeneous neuronal dynamics without relying on backpropagation?

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.

In response to this challenge, we note that Reward-driven connectivity optimization can adapt binary connectivity without relying on backpropagation. Evolving Connectivity (EC), for example, learns binary connectivity through a Bernoulli parameterization and achieves performance comparable to deep recurrent neural networks(Wang et al., 2023). This demonstrates that Reward can provide an effective task-level signal for structural adaptation. Joint adaptation further requires the shared Reward to be associated with both structural and dynamical parameter variations. Forward-gradient optimization provides such optimization information through directional estimates obtained by forward computation(Silver et al., 2022; Fournier et al., 2023). These estimates can be accumulated across successive optimization steps to approximate the underlying gradient. This motivates a directional credit formulation for jointly adapting connectivity structure and neuronal dynamics.

To this end, we propose Directional Credit guided Forward Optimization (DCFO), a biologically inspired learning framework for RSNNs. DCFO jointly adapts connectivity structure and neuron-specific membrane time constants without relying on backpropagation. Candidate directions are constructed separately in the connectivity parameter space and the neuron-specific membrane-time-constant space. The shared reward is then used to evaluate the candidate directions in each space. This yields directional credit for connectivity structure and neuronal dynamics, respectively. Together, these updates enable structural–dynamical co-adaptation.

To assess its effectiveness, we systematically evaluate DCFO across a set of widely used sequential decision-making tasks, including the 17-DoF Humanoid(Brockman et al., 2016; Freeman et al., 2021). The results show that jointly optimizing connectivity structure and neuronal dynamics improves task performance over the connectivity-optimization baseline and achieves performance comparable to deep recurrent neural networks. The learned neuronal dynamics further exhibit task-dependent heterogeneous neuronal timescales.

The main contributions of this work are summarized as follows:

- We propose a backpropagation-free learning framework, termed Directional Credit guided Forward Optimization (DCFO), which assigns Directional Credit to candidate directions and uses the resulting credits to form forward-gradient updates for RSNN optimization.
- We further introduce a biologically inspired joint adaptation mechanism that enables coarse-grained reconfiguration of connectivity and fine-grained adjustment of neuron-specific membrane time constants within a unified forward-optimization framework.
- We systematically evaluate DCFO on three Brax continuous-control tasks, including the 17-DoF Humanoid, demonstrating the effectiveness of the proposed structural–dynamical co-adaptation mechanism.

2 Related Work

2.1 Learnable Neuronal Dynamics

Recent studies have increasingly focused on making neuronal dynamics trainable. The Parametric Leaky Integrate-and-Fir (PLIF) systematically parameterizes the membrane time constant as a learnable variable and jointly optimizes it with synaptic weights (Fang et al., 2021). Multi-timescale adaptive RSNNs further learn both membrane and adaptation time constants (Yin et al., 2020; 2021). Subsequent work extends optimization from a single neuronal parameter to multiple membrane-related dynamical properties (Rathi & Roy, 2023; Yao et al., 2022; Kosta & Roy, 2023; Hao et al., 2024; Wang et al., 2022; Ding et al., 2022; Xu et al., 2024; Bai et al., 2026). Finer-grained heterogeneous models further extend temporal adaptation from whole-neuron parameters to dendritic compartments and individual synapses (Zheng et al., 2024; Deng et al., 2025). These studies establish heterogeneous neuronal dynamics as an effective basis for representing diverse temporal dependencies(Perez-Nieves et al., 2021; Chakraborty & Mukhopadhyay, 2023). However, existing approaches to learning neuronal dynamics rely predominantly on surrogate-gradient backpropagation(Neftci et al., 2019). Gradient-based optimization therefore faces a practical limitation: the learnability of neuronal dynamics has not yet been fully translated into adaptability on neuromorphic hardware. Forward Propagation Through Time with Liquid Time-Constant spiking neurons (FPTT-LTC) partially alleviates this limitation by supporting online learning of neuronal dynamics over long sequences through forward propagation through time, but it only removes backward

temporal unrolling and still relies on surrogate gradients and differentiable per-timestep losses (Yin et al., 2023). We therefore introduce a backpropagation-free framework for learning heterogeneous, neuron-specific membrane time constants.

2.2 Backpropagation-Free Connectivity Optimization

Connectivity optimization treats the connectivity of a neural network as an optimization variable and adjusts the network structure to form more effective information-transmission pathways. Conventional connectivity optimization is typically coupled with gradient-driven parameter training, progressively forming sparse and effective network structures (Han et al., 2015; Louizos et al., 2018; Bellec et al., 2018a; Evci et al., 2020). This idea was subsequently extended to SNNs, allowing connectivity structures to be jointly adjusted with spiking-network parameters during training (Chen et al., 2021; 2022; Shen et al., 2025). However, these structure-learning methods still rely on backpropagation.

To remove this dependence, Weight Agnostic Neural Networks show that network topology itself can carry task-relevant information (Gaier & Ha, 2019), motivating direct search over connectivity structures based on task evaluation. Evolving Connectivity (EC) further extends this idea to RSNNs, enabling backpropagation-free optimization of 1-bit recurrent connectivity structures (Wang et al., 2023). However, its optimization process does not explicitly distinguish the local contributions of different structural changes. Building on the existing 1-bit connectivity optimization framework of Evolving Connectivity, we further extend connectivity optimization through Directional Credit guided Forward Optimization.

2.3 Forward-Gradient Optimization

Conventional neural network optimization typically relies on backpropagation to compute gradients. Directional Gradient Descent established a foundation for forward-gradient optimization, showing that directional derivatives of the objective along candidate directions can provide effective update signals (Silver et al., 2022). It therefore provides a viable route to parameter optimization without backpropagation. Subsequent studies further show that local objective design and candidate-direction quality are critical to the efficiency of forward-gradient optimization (Ren et al., 2023; Fournier et al., 2023). Meanwhile, randomized forward-mode gradients have been shown to optimize continuous synaptic weights in SNNs without backpropagation (Wan et al., 2026).

However, existing forward-gradient methods mainly target continuous parameters and differentiable objectives, making them difficult to directly apply to task-return-driven optimization of discrete connectivity(Silver et al., 2022; Ren et al., 2023; Fournier et al., 2023; Wan et al., 2026). Building on these advances, we construct directional credit from sampled returns and extend forward-gradient optimization to the joint optimization of RSNN connectivity and neuronal dynamics. This opens new possibilities for hardware-friendly forward optimization of recurrent spiking neural networks.

3 Preliminaries

3.1 Recurrent Spiking Neural Network

We consider an LIF-based recurrent spiking neural network (RSNN) that processes temporal information through recurrent connectivity and spiking dynamics(Bellec et al., 2018b). Recurrent connectivity determines the pathways through which information propagates across the network, whereas neuronal dynamics govern the timescales of temporal integration and decay.

Each recurrent neuron is modeled as a leaky integrate-and-fire (LIF) neuron(Burkitt, 2006). For neuron h, the membrane time constant τ_h determines the membrane decay factor (Fang et al., 2021)

$$\beta_h = \exp\left(-\frac{\Delta t}{\tau_h}\right), \tag{1}$$

where Δt is the simulation time interval. The decay factor controls how strongly the membrane state retains past inputs and therefore determines the neuron's temporal integration and decay timescale.

We allow the membrane time constant to vary across neurons and denote

$$\tau = [\tau_1, \ldots, \tau_H]^\top, \tag{2}$$

where H is the number of neurons with learnable membrane time constants. This neuron-specific parameterization allows the RSNN to express heterogeneous neuronal dynamics through different temporal integration timescales (Perez-Nieves et al., 2021).

3.2 Structural and Dynamical Parameterization

We represent connectivity structure and neuronal dynamics by two parameter blocks, corresponding to Bernoulli connectivity parameters and neuron-specific membrane time constants, respectively.

For structural adaptation, let D denote the number of candidate recurrent connections. At optimization iteration t, the connectivity structure is parameterized by $\rho_t = [\rho_{t,1}, \ldots, \rho_{t,D}]^\top \in [0,1]^D$. The i-th sampled RSNN instantiates a binary connectivity configuration $\theta_{i,t} \in \{0,1\}^D$, where $\theta_{i,t,j} = 1$ indicates that connection j is active. Each connection state is sampled independently as $\theta_{i,t,j} \sim \mathrm{Bernoulli}(\rho_{t,j})$(Wang et al., 2023), yielding

$$q_{\rho_t}(\theta_{i,t}) = \prod_{j=1}^{D} \rho_{t,j}^{\theta_{i,t,j}} (1-\rho_{t,j})^{1-\theta_{i,t,j}}. \tag{3}$$

Thus, ρ_t parameterizes the adaptive connectivity structure, whereas $\theta_{i,t}$ specifies the binary connectivity instantiated for one RSNN evaluation.

For dynamical adaptation, let $\tau_t = [\tau_{t,1}, \ldots, \tau_{t,H}]^\top \in \mathcal{T}$ denote the current neuron-specific membrane time constants. To sample local dynamical variations around τ_t, we draw a ternary perturbation vector $\mathbf{s}_{i,t}$ and construct bounded antithetic samples(Salimans et al., 2017; Maheswaranathan et al., 2019)

$$\tilde{\tau}_{i,t}^{\pm} = \Pi_{\mathcal{T}}\left(\tau_t \pm \epsilon_\tau \mathbf{s}_{i,t}\right), \qquad s_{i,t,h} \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}\{-1,0,+1\}. \tag{4}$$

Here, $\epsilon_\tau > 0$ is the perturbation magnitude, $\mathcal{T}$ is the feasible set of membrane time constants, and $\Pi_{\mathcal{T}}$ denotes projection onto this set.

4 Method

In this section, we present Directional Credit-guided Forward Optimization (DCFO), as illustrated in Fig. 1. DCFO constructs candidate directions separately for connectivity structure and neuron-specific membrane time constants. Each candidate direction is evaluated by its local Directional Credit, and the resulting credits are jointly reconstructed into forward-gradient update directions for the two parameter blocks. These coordinated updates enable structural–dynamical co-adaptation, allowing the RSNN to develop task-adaptive connectivity structure and heterogeneous neuronal dynamics governed by neuron-specific membrane time constants.

4.1 Joint Sampling under a Shared reward Objective

DCFO instantiates structural and dynamical variations using the sampling schemes defined in equation 3 and equation 4. Specifically, binary connectivity configurations $\theta_{i,t}$ are sampled from q_{ρ_t}, while paired membrane-time-constant samples $\tilde{\tau}_{i,t}^{\pm}$ are constructed around the current τ_t.

Let $R(\theta, \tau)$ denote the reward obtained by an RSNN instantiated with connectivity θ and membrane time constants τ. Under these structural and dynamical sampling distributions, we optimize the perturbation-smoothed expected reward

$$\max_{\substack{\rho \in [0,1]^D \\ \tau \in \mathcal{T}}} \mathcal{J}_{\epsilon_\tau}(\rho, \tau) = \mathbb{E}_{\substack{\theta \sim q_\rho \\ \mathbf{s} \sim \mathrm{Unif}(\{-1,0,+1\}^H)}} \left[R(\theta, \Pi_{\mathcal{T}}(\tau + \epsilon_\tau \mathbf{s}))\right]. \tag{5}$$

This objective couples connectivity structure and neuronal dynamics through the same task-level reward. Since the perturbation distribution is symmetric, the objective can be written using a single perturbation sign, while both antithetic evaluations are retained in practice to construct the dynamical evidence described below.

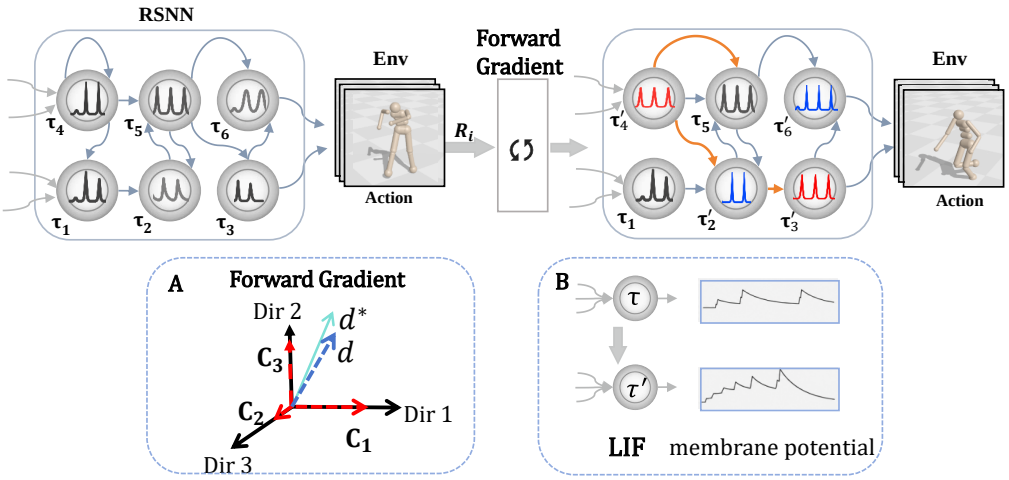

Figure 1: Overview of DCFO for jointly adapting recurrent connectivity structure and neuronal dynamics in an RSNN with heterogeneous membrane time constants ($\tau_1 \neq \tau_2 \neq \cdots \neq \tau_6$). **A**, the Forward Gradient module evaluates candidate parameter-change directions, computes the corresponding directional credit, and forms the forward update. **B**, changes in membrane time constants reshape LIF membrane-potential dynamics. Red and blue indicate increased and decreased membrane time constants, respectively.

4.2 Directional Credit Assignment and Reconstruction

As illustrated by the forward-gradient module in Fig. 1, DCFO treats candidate directions as the basic units of Directional Credit assignment. Candidate directions are constructed from task-related parameter variations available during optimization, with the specific construction used in our experiments described in Sec. 5. For compact notation, let $b \in \{\rho, \tau\}$ denote the parameter block, with

$$x_t^\rho = \rho_t, \qquad x_t^\tau = \tau_t.$$

For block b, let $v_{t,k}^b$, $k = 1, \ldots, K_b$, denote the k-th normalized candidate direction, with $\|v_{t,k}^b\|_2 = 1$. The candidate directions are collected in

$$\mathbf{V}_t^b = \begin{bmatrix} v_{t,1}^b & \cdots & v_{t,K_b}^b \end{bmatrix}^\top. \tag{6}$$

Each candidate specifies a local parameter variation whose directional support is evaluated through the corresponding credit-evaluation function.

DCFO evaluates each candidate direction through a forward-mode Jacobian–vector product (JVP). A JVP(Baydin et al., 2018; Silver et al., 2022; Wan et al., 2026). measures the first-order local response of a function along a specified vector direction. For the scalar credit-evaluation function $\mathcal{E}_t^b$, it is given by $\mathrm{JVP}(\mathcal{E}_t^b, x_t^b; v_{t,k}^b) = \nabla_{x^b} \mathcal{E}_t^b(x_t^b)^\top v_{t,k}^b$, which directly evaluates the directional derivative without explicitly forming the full gradient.

DCFO uses the negative directional response as the directional credit:

$$\widehat{c}_{t,k}^b = -\,\mathrm{JVP}\left(\mathcal{E}_t^b, x_t^b; v_{t,k}^b\right). \tag{7}$$

A positive $\widehat{c}_{t,k}^b$ indicates that a local move along $v_{t,k}^b$ decreases the current credit-evaluation function to first order, while its magnitude quantifies the strength of the directional support.

During JVP evaluation, the sampled reward evidence is treated as fixed. Differentiation is performed only through $\mathcal{E}_t^b$, without differentiating through the RSNN rollout or environment trajectory. We collect the resulting directional credits as $\widehat{\mathbf{c}}_t^b = [\widehat{c}_{t,1}^b, \ldots, \widehat{c}_{t,K_b}^b]^\top$.

Because candidate directions can be correlated, their directional credits are reconstructed jointly rather than combined independently. We obtain the reconstruction coefficients α_t^b by solving

$$\left[\mathbf{V}_t^b (\mathbf{V}_t^b)^\top + \lambda_b I_{K_b}\right] \alpha_t^b = \widehat{\mathbf{c}}_t^b, \qquad d_t^b = (\mathbf{V}_t^b)^\top \alpha_t^b. \tag{8}$$

Here, $\lambda_b > 0$ regularizes the reconstruction(Hoerl & Kennard, 1970), and d_t^b is the reconstructed update direction for block b. This reconstruction maps the correlated directional credits back to a coordinated parameter-space update direction.

Algorithm 1 Directional Credit guided Forward Optimization(DCFO)

Require: Initial connectivity parameters ρ_0, membrane time constants τ_0, perturbation scale ϵ_τ, regularization coefficients $\lambda_\rho, \lambda_\tau$, learning rates η_ρ, η_τ

for $t = 0, 1, \ldots, T-1$ **do**
  Sample connectivity configurations $\theta_{i,t} \sim q_{\rho_t}$
  Sample ternary perturbations $\mathbf{s}_{i,t}$ and construct $\tilde{\tau}^{\pm}_{i,t} = \Pi_{\mathcal{T}}(\tau_t \pm \epsilon_\tau \mathbf{s}_{i,t})$
  Evaluate paired rewards $R^{\pm}_{i,t} = R(\theta_{i,t}, \tilde{\tau}^{\pm}_{i,t})$ under the same connectivity $\theta_{i,t}$
  Construct structural evidence $\xi^{\rho}_{i,t} = \theta_{i,t} - \rho_t$ and centered reward-derived weights $w^{\rho}_{i,t}$
  Construct dynamical evidence $\Delta R^{\tau}_{i,t} = R^{+}_{i,t} - R^{-}_{i,t}$ and $\xi^{\tau}_{i,t} = \tilde{\tau}^{+}_{i,t} - \tilde{\tau}^{-}_{i,t}$
  Construct normalized candidate matrices $\mathbf{V}^{\rho}_t$ and $\mathbf{V}^{\tau}_t$
  for $b \in \{\rho, \tau\}$ **do**
    Compute the directional credit vector $\widehat{\mathbf{c}}^b_t$, where $\widehat{c}^b_{t,k} = -\mathrm{JVP}(\mathcal{E}^b_t, x^b_t; v^b_{t,k})$, $k = 1, \ldots, K_b$
    Solve $\left[\mathbf{V}^b_t (\mathbf{V}^b_t)^\top + \lambda_b I_{K_b}\right] \alpha^b_t = \widehat{\mathbf{c}}^b_t$
    Reconstruct $d^b_t = (\mathbf{V}^b_t)^\top \alpha^b_t$
  end for
  Update connectivity parameters $\rho_{t+1} = \Pi_{[0,1]^D}(\rho_t + \eta_\rho d^{\rho}_t)$
  Update membrane time constants $\tau_{t+1} = \Pi_{\mathcal{T}}(\tau_t + \eta_\tau d^{\tau}_t)$
end for

Proposition 1 (Local descent of the reconstructed direction)**.** *Assume that $\mathcal{E}^b_t$ is differentiable at x^b_t and that $\lambda_b > 0$. If $\widehat{\mathbf{c}}^b_t \neq \mathbf{0}$, then the reconstructed direction d^b_t satisfies*

$$\left\langle \nabla_{x^b} \mathcal{E}^b_t(x^b_t), d^b_t \right\rangle = -\|d^b_t\|_2^2 - \lambda_b \|\alpha^b_t\|_2^2 < 0. \tag{9}$$

Hence, d^b_t is a strict local descent direction of the current credit-evaluation function. The proof is provided in Appendix B.

4.3 Block-Specific Credit Construction

The two parameter blocks differ in their parameterization and local variation mechanisms. Connectivity structure is instantiated through binary, discrete connectivity configurations, whereas neuron-specific membrane time constants are continuous parameters. Accordingly, DCFO uses block-specific sampling and credit construction to associate the shared reward with structural and dynamical parameter variations.

For connectivity structure, define the sampled structural variation as $\xi^{\rho}_{i,t} = \theta_{i,t} - \rho_t$, and let $w^{\rho}_{i,t}$ denote the reward-derived structural weight associated with $\theta_{i,t}$. The structural credit assigned to candidate direction $v^{\rho}_{t,k}$ is

$$\widehat{c}^{\rho}_{t,k} = \frac{1}{N} \sum_{i=1}^{N} w^{\rho}_{i,t} \left\langle \xi^{\rho}_{i,t}, v^{\rho}_{t,k} \right\rangle. \tag{10}$$

This credit measures the alignment between a candidate direction and reward-weighted connectivity variations. Because these variations arise from sampled binary connectivity configurations, they provide a coarse-grained mechanism for reconfiguring recurrent information-flow pathways.

For neuronal dynamics, the antithetic membrane-time-constant samples are evaluated under the same sampled connectivity $\theta_{i,t}$, yielding

$$R^{\pm}_{i,t} = R(\theta_{i,t}, \tilde{\tau}^{\pm}_{i,t}), \qquad \Delta R^{\tau}_{i,t} = R^{+}_{i,t} - R^{-}_{i,t}.$$

Because the paired evaluations share the same structural context, $\Delta R^{\tau}_{i,t}$ provides a structure-conditioned signal for the corresponding dynamical variation and reduces cross-block credit ambiguity. Defining $\xi^{\tau}_{i,t} = \tilde{\tau}^{+}_{i,t} - \tilde{\tau}^{-}_{i,t}$, the dynamical credit assigned to candidate direction $v^{\tau}_{t,k}$ is

$$\widehat{c}^{\tau}_{t,k} = \frac{1}{M} \sum_{i=1}^{M} \Delta R^{\tau}_{i,t} \left\langle \xi^{\tau}_{i,t}, v^{\tau}_{t,k} \right\rangle. \tag{11}$$

Since these variations are local perturbations in the continuous membrane-time-constant space, the resulting credit supports fine-grained adjustment of neuronal temporal integration and decay.

The resulting structural and dynamical credit vectors are processed by the shared reconstruction in equation 8, yielding the reconstructed update directions d_t^ρ and d_t^τ. In this way, DCFO coordinates coarse-grained reconfiguration of connectivity structure with fine-grained adaptation of neuronal dynamics under the same reward objective, enabling structural–dynamical co-adaptation without relying on backpropagation. Algorithm 1 summarizes the resulting joint optimization procedure.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUPS

Tasks. We evaluate our method on three robotic locomotion benchmarks widely used in reinforcement learning: Humanoid, Walker2d, and Hopper (Freeman et al., 2021). As illustrated in Fig. 1 for the 17-degree-of-freedom (DoF) Humanoid, the RSNN maps environmental observations to continuous control actions to maximize cumulative reward over an episode of length L. For the policy $\pi_{\theta,\tau}$ specified by binary connectivity θ and the membrane time-constant vector τ, performance is defined as the expected return over episodes: $R(\theta,\tau) = \mathbb{E}_{\xi\sim p(\cdot|\pi_{\theta,\tau})}\left[\sum_{\ell=0}^{L-1} r_\ell\right]$, A single evaluation R_i corresponds to the cumulative return of one episode using the sampled configuration $(\theta_i, \tilde{\tau}_i)$.

Baselines. We compare DCFO with representative RSNN and recurrent-network baselines following the evaluation setting of EC (Wang et al., 2023). EC-RSNN is the primary connectivity-optimization baseline, which uses NES to optimize Bernoulli-parameterized binary connectivity from reward while keeping membrane time constants fixed(Wang et al., 2023; Wierstra et al., 2014). ES-RSNN directly evolves continuous RSNN weights using Evolution Strategies (ES) (Salimans et al., 2017; Wang et al., 2023). We further include ES-GRU and ES-LSTM, where conventional GRU(Cho et al., 2014) and LSTM (Hochreiter & Schmidhuber, 1997) policies are optimized with ES(Salimans et al., 2017).

Implementation Details. Our framework and all baselines are implemented using JAX (Bradbury et al., 2018) and just-in-time compiled for efficient GPU execution. Brax (Freeman et al., 2021) is used as the physics simulator for all continuous-control tasks. Each reported result is averaged over three independent seeds, with the standard deviation shown as a shaded region. Detailed hyperparameters and hardware specifications are provided in Appendix A.

5.2 PERFORMANCE EVALUATION

Connectivity-Optimization Comparison. We first compare DCFO and EC using the same RSNN architecture and population size. We evaluate DCFO (Structure Only), which fixes τ and optimizes only ρ, and DCFO with joint optimization of connectivity structure and membrane time constants. As shown in Fig. 2 and Table 1, joint optimization of connectivity structure and membrane time constants achieves higher mean final Reward than EC on all three tasks and further improves over DCFO (Structure Only) on Humanoid and Walker2d, while remaining comparable on Hopper. Notably, joint optimization achieves particularly strong performance on the most complex 17-DoF Humanoid task. In terms of generation efficiency, Table 2 shows that joint optimization of connectivity structure and membrane time constants reaches the predefined target Reward in fewer generations than EC on all three tasks.

Comparison with ES-Trained Recurrent Networks. Beyond the connectivity-optimization baseline, we further compare DCFO with ES-RSNN, ES-GRU, and ES-LSTM to evaluate its performance against recurrent networks optimized through direct continuous-parameter evolution. ES-RSNN applies Evolution Strategies directly to the continuous weights of an RSNN, while ES-GRU and ES-LSTM provide conventional deep recurrent-network baselines. As shown in Table 1, DCFO achieves higher mean final returns than ES-RSNN on all three tasks, showing that strong task performance can be achieved without directly evolving continuous RSNN weights. Compared with the deep recurrent baselines, DCFO achieves higher mean final returns than both ES-GRU and ES-LSTM on Walker2d and Hopper.

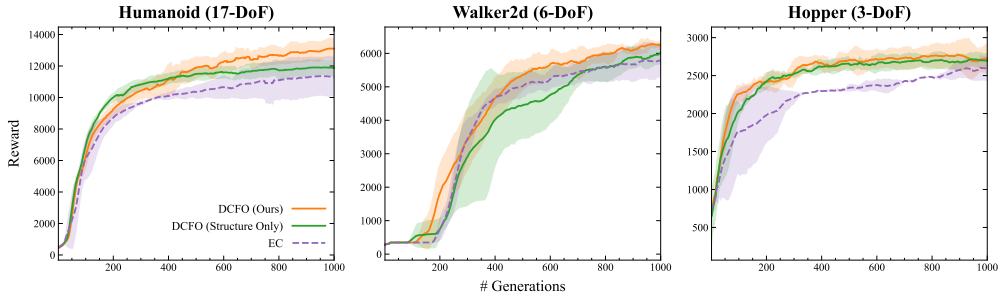

Figure 2: Performance comparison under connectivity-structure optimization. All methods optimize the RSNN connectivity structure, while differing in whether and how neuronal dynamics are adapted. The figure compares full DCFO, DCFO (Structure Only), and EC on Humanoid, Walker2d, and Hopper.

Table 1: Mean final evaluation returns and sample standard deviations.

	Task		
Method	**Humanoid (17-DoF)**	**Walker2d (6-DoF)**	**Hopper (3-DoF)**
DCFO (Ours)	13,104.31 (±680.56)	**6,244.51** (±149.32)	**2,725.81** (±208.47)
DCFO (Structure Only)	11,865.84 (±533.92)	6,024.31 (±313.79)	2,695.70 (±92.14)
EC-RSNN	11,316.62 (±1,271.69)	5,775.98 (±481.86)	2,404.05 (±340.22)
ES-RSNN	11,260.42 (±1,184.37)	3,912.76 (±968.54)	2,604.38 (±96.27)
ES-GRU	13,085.67 (±1,842.53)	4,718.63 (±1,327.46)	1,702.16 (±1,248.75)
ES-LSTM	**14,782.36** (±438.91)	4,328.51 (±572.83)	2,184.72 (±836.41)

On the more complex 17-DoF Humanoid task, DCFO achieves a mean final return comparable to ES-GRU, although ES-LSTM attains the highest return. Taken together, these results show that RSNNs optimized with DCFO achieve competitive performance with ES-trained deep recurrent networks.

Analysis of Performance Differences. We next analyze the performance differences from two complementary perspectives: how candidate directions constrain structural updates, and how adaptive membrane time constants provide fine-grained optimization of neuronal dynamics.

First, DCFO (Structure Only) achieves higher mean final returns than EC on all three tasks. EC updates the connectivity distribution from configuration-level Reward, so each scalar Reward reflects many simultaneous connectivity changes without explicitly separating their directional contributions (Wang et al., 2023). In contrast, DCFO constructs structural candidate directions from task-related connectivity variations, including high-Reward structural changes, differences between high- and low-Reward samples, and the previous structural update. It then assigns directional credit to these candidates and reconstructs the connectivity update within their candidate subspace. This explicitly separates directional contributions and constrains the update to task-relevant structural variations, improving the quality of structural optimization. The candidate-count ablation in Appendix D further shows that performance generally improves with directional coverage, with $K_\rho = 4096$ achieving the highest final Reward on Humanoid within the tested range. These results indicate that effective structural optimization requires both informative candidate directions and sufficient coverage of the structural search space.

Second, membrane time constants provide a fine-grained degree of freedom for neuronal dynamics by controlling neuron-specific temporal integration and decay (Fang et al., 2021; Perez-Nieves et al., 2021). We compare fixed membrane time constants of $\tau = 5$ and $\tau = 15$ while keeping structural optimization unchanged. As shown in Fig. 3a, $\tau = 15$ achieves higher final Reward on Humanoid for both DCFO (Structure Only) and EC, whereas Fig. 3b shows that $\tau = 5$ improves Reward more rapidly during early training on Walker2d. These results show that membrane timescales substantially affect structural optimization, with task-dependent effects.

We further compare fixed heterogeneous membrane time constants sampled from a uniform distribution with joint optimization of connectivity structure and neuronal dynamics. Both settings use

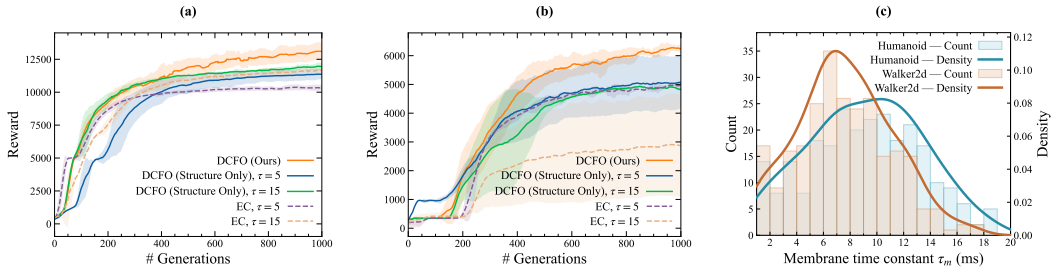

Figure 3: (a,b) Performance of connectivity-only DCFO and EC under different fixed membrane time constants. (c) Learned membrane-time-constant distributions after joint optimization on Humanoid and Walker2d.

Table 2: Generations and wall-clock time required to reach the target return. N.R. denotes that the target return was not reached.

	Humanoid (17-DoF) Target: 11,000		Walker2d (6-DoF) Target: 5,500		Hopper (3-DoF) Target: 2,500	
Method	Gen.	Time (min)	Gen.	Time (min)	Gen.	Time (min)
DCFO (Ours)	390	367.18	**534**	324.32	290	159.78
DCFO (Structure Only)	346	360.93	731	332.25	260	164.22
EC-RSNN	725	352.45	752	297.52	837	384.13
ES-RSNN	686	1887.30	N.R.	N.R.	**159**	410.68
ES-GRU	267	157.15	N.R.	N.R.	N.R.	N.R.
ES-LSTM	**194**	83.85	N.R.	N.R.	N.R.	N.R.

the same uniformly sampled neuron-specific initialization, while only full DCFO updates τ during training. As shown in Appendix E, full DCFO achieves higher Reward than fixed heterogeneity on Humanoid. Although fixed heterogeneity provides diverse neuronal timescales (Perez-Nieves et al., 2021), their allocation cannot adapt to task feedback. Joint optimization instead allows neuron-specific temporal integration and decay to adapt with connectivity structure. Consistently, Fig. 3c shows that DCFO reshapes the initially uniform membrane-time-constant distribution into a nonuniform distribution, indicating adaptive neuron-specific timescale allocation rather than fixed random heterogeneity.

Together, candidate directions constrain structural updates, while adaptive membrane time constants provide fine-grained optimization of neuronal dynamics, enabling connectivity structure and neuronal dynamics to co-adapt to task demands.To further verify the effectiveness of Directional Credit guided Forward Optimization for membrane-time-constant optimization, we compare Tau-DCFO with an ES-based alternative in Appendix C.

6 Conclusion

In this work, we propose Directional Credit guided Forward Optimization (DCFO), a biologically inspired, backpropagation-free learning framework for RSNNs. DCFO exploits directional credit along candidate parameter-change directions to jointly optimize connectivity structure and membrane time constants within a unified forward-optimization framework. Although a trade-off remains between candidate-direction coverage and computational cost, the backpropagation-free nature of DCFO makes it suitable for training RSNNs on neuromorphic devices. More broadly, learning in biological neural systems involves multi-level adaptation of network connectivity and intrinsic neuronal dynamics. Developing neural models whose learning mechanisms and adaptation of neuronal dynamics more closely reflect neurophysiological processes remains an important direction for future research.

AI USE STATEMENT

Generative AI tools were used for language editing and literature search. All scientific claims, mathematical formulations, experimental results, and conclusions were reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

REFERENCES

Qianyi Bai, Haiteng Wang, and Qiang Yu. A brain-inspired gating mechanism unlocks robust computation in spiking neural networks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/fc92064236f06645e0b5ec5a8a16565d-Abstract-Conference.html.

Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018. URL https://www.jmlr.org/papers/v18/17-468.html.

Guillaume Bellec, David Kappel, Wolfgang Maass, and Robert Legenstein. Deep rewiring: Training very sparse deep networks. In *International Conference on Learning Representations*, 2018a. URL https://openreview.net/forum?id=BJ_wN01C-.

Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. In *Advances in Neural Information Processing Systems*, volume 31, 2018b. URL https://proceedings.neurips.cc/paper/2018/hash/c203d8a151612acf12457e4d67635a95-Abstract.html.

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. doi: 10.1038/s41467-020-17236-y. URL https://www.nature.com/articles/s41467-020-17236-y.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: Composable transformations of Python+NumPy programs, 2018. URL https://github.com/jax-ml/jax. Software.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Anthony N. Burkitt. A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological Cybernetics*, 95(1):1–19, 2006. doi: 10.1007/s00422-006-0068-6.

Biswadeep Chakraborty and Saibal Mukhopadhyay. Heterogeneous neuronal and synaptic dynamics for spike-efficient unsupervised learning: Theory and design principles. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=QIRtAqoXwj.

Yanqi Chen, Zhaofei Yu, Wei Fang, Tiejun Huang, and Yonghong Tian. Pruning of deep spiking neural networks through gradient rewiring. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 1713–1721. International Joint Conferences on Artificial Intelligence Organization, 2021. doi: 10.24963/ijcai.2021/236. URL https://www.ijcai.org/proceedings/2021/236.

Yanqi Chen, Zhaofei Yu, Wei Fang, Zhengyu Ma, Tiejun Huang, and Yonghong Tian. State transition of dendritic spines improves learning of sparse spiking neural networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3701–3715. PMLR, 2022. URL https://proceedings.mlr.press/v162/chen22ac.html.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179/.

Zhichao Deng, Zhikun Liu, Junxue Wang, Shengqian Chen, Xiang Wei, and Qiang Yu. HetSyn: Versatile timescale integration in spiking neural networks via heterogeneous synapses. In *Advances in Neural Information Processing Systems*, volume 38, pp. 45817–45848, 2025. doi: 10.52202/085713-1367. URL https://papers.nips.cc/paper_files/paper/2025/hash/3a4e2c6a07ede771fbe53201d31df1e9-Abstract-Conference.html.

Jianchuan Ding, Bo Dong, Felix Heide, Yufei Ding, Yunduo Zhou, Baocai Yin, and Xin Yang. Biologically inspired dynamic thresholds for spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 6090–6103, 2022. doi: 10.52202/068431-0441. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/2858f8c8683aaa8c12d487354cf328dc-Abstract-Conference.html.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2943–2952. PMLR, 2020. URL https://proceedings.mlr.press/v119/evci20a.html.

Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2661–2671, October 2021. doi: 10.1109/ICCV48922.2021.00266. URL https://openaccess.thecvf.com/content/ICCV2021/html/Fang_Incorporating_Learnable_Membrane_Time_Constant_To_Enhance_Learning_of_Spiking_ICCV_2021_paper.html.

Louis Fournier, Stephane Rivaud, Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Can forward gradient match backpropagation? In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10249–10264. PMLR, 2023. URL https://proceedings.mlr.press/v202/fournier23a.html.

C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax - a differentiable physics engine for large scale rigid body simulation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/d1f491a404d6854880943e5c3cd9ca25-Abstract-round1.html.

Adam Gaier and David Ha. Weight agnostic neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/e98741479a7b998f88b8f8c9f0b6b6f1-Abstract.html.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, volume 28, pp. 1135–1143, 2015.

Zecheng Hao, Xinyu Shi, Zihan Huang, Tong Bu, Zhaofei Yu, and Tiejun Huang. A progressive training framework for spiking neural networks with learnable multi-hierarchical model. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 4925–4941, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/148865706acbd18627d3fc15cc3f3b93-Paper-Conference.pdf.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

Adarsh Kumar Kosta and Kaushik Roy. Adaptive-SpikeNet: Event-based optical flow estimation using spiking neural networks with learnable neuronal dynamics. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6021–6027, 2023. doi: 10.1109/ICRA48891.2023.10160551. URL https://ieeexplore.ieee.org/document/10160551.

Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through L_0 regularization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1Y8hhg0b.

Niru Maheswaranathan, Luke Metz, George Tucker, Dami Choi, and Jascha Sohl-Dickstein. Guided evolutionary strategies: Augmenting random search with surrogate gradients. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4264–4273. PMLR, 2019. URL https://proceedings.mlr.press/v97/maheswaranathan19a.html.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019. doi: 10.1109/MSP.2019.2931595. URL https://ieeexplore.ieee.org/document/8891809.

Nicolas Perez-Nieves, Vincent C. H. Leung, Pier Luigi Dragotti, and Dan F. M. Goodman. Neural heterogeneity promotes robust learning. *Nature Communications*, 12(1):5791, 2021. doi: 10.1038/s41467-021-26022-3. URL https://www.nature.com/articles/s41467-021-26022-3.

Nitin Rathi and Kaushik Roy. DIET-SNN: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 34(6):3174–3182, 2023. doi: 10.1109/TNNLS.2021.3111897. URL https://ieeexplore.ieee.org/document/9556508.

Mengye Ren, Simon Kornblith, Renjie Liao, and Geoffrey Hinton. Scaling forward gradient with local losses. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JxpBP1JM15-.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017. URL https://arxiv.org/abs/1703.03864.

Jiangrong Shen, Qi Xu, Gang Pan, and Badong Chen. Improving the sparse structure learning of spiking neural networks from the view of compression efficiency. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e7fd2c0a1a6f956c94024e955b34cc43-Abstract-Conference.html.

David Silver, Anirudh Goyal, Ivo Danihelka, Matteo Hessel, and Hado van Hasselt. Learning by directional gradient descent. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=5i7lJLuhTm.

Ruyin Wan, Qian Zhang, and George Em Karniadakis. Randomized forward mode gradient for spiking neural networks in scientific machine learning. *Neural Networks*, 201:108978, 2026. doi: 10.1016/j.neunet.2026.108978. URL https://www.sciencedirect.com/science/article/pii/S0893608026004399.

Guan Wang, Yuhao Sun, Sijie Cheng, and Sen Song. Evolving connectivity for recurrent spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 2991–3007, 2023. doi: 10.52202/075280-0133. URL

https://proceedings.neurips.cc/paper_files/paper/2023/hash/08f9de0232c0b485110237f6e6cf88f1-Abstract-Conference.html.

Siqi Wang, Tee Hiang Cheng, and Meng-Hiot Lim. LTMD: Learning improvement of spiking neural networks with learnable thresholding neurons and moderate dropout. In *Advances in Neural Information Processing Systems*, volume 35, pp. 28350–28362, 2022. doi: 10.52202/068431-2055. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b5fd95d6b16d3172e307103a97f19e1b-Abstract-Conference.html.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *Journal of Machine Learning Research*, 15(27):949–980, 2014. URL https://jmlr.org/papers/v15/wierstra14a.html.

Mengting Xu, De Ma, Huajin Tang, Qian Zheng, and Gang Pan. FEEL-SNN: Robust spiking neural networks with frequency encoding and evolutionary leak factor. In *Advances in Neural Information Processing Systems*, volume 37, pp. 91930–91950, 2024. doi: 10.52202/079017-2917. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a73474c359ed523e6cd3174ed29a4d56-Abstract-Conference.html.

Xingting Yao, Fanrong Li, Zitao Mo, and Jian Cheng. GLIF: A unified gated leaky integrate-and-fire neuron for spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 32160–32171, 2022. doi: 10.52202/068431-2330. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/cfa8440d500a6a6867157dfd4eaff66e-Abstract-Conference.html.

Bojian Yin, Federico Corradi, and Sander M. Bohté. Effective and efficient computation with multiple-timescale spiking recurrent neural networks. In *International Conference on Neuromorphic Systems 2020*, pp. 1–8, 2020. doi: 10.1145/3407197.3407225. URL https://dl.acm.org/doi/10.1145/3407197.3407225.

Bojian Yin, Federico Corradi, and Sander M. Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10): 905–913, 2021. doi: 10.1038/s42256-021-00397-w. URL https://www.nature.com/articles/s42256-021-00397-w.

Bojian Yin, Federico Corradi, and Sander M. Bohté. Accurate online training of dynamical spiking neural networks through forward propagation through time. *Nature Machine Intelligence*, 5(5): 518–527, 2023. doi: 10.1038/s42256-023-00650-4. URL https://www.nature.com/articles/s42256-023-00650-4.

Hanle Zheng, Zhong Zheng, Rui Hu, Bo Xiao, Yujie Wu, Fangwen Yu, Xue Liu, Guoqi Li, and Lei Deng. Temporal dendritic heterogeneity incorporated with spiking neural networks for learning multi-timescale dynamics. *Nature Communications*, 15(1):277, 2024. doi: 10.1038/s41467-023-44614-z. URL https://www.nature.com/articles/s41467-023-44614-z.

A Detailed Experiment Settings

All experiments were conducted on a single-GPU workstation with the following specifications. All efficiency experiments, including those measuring wall-clock time, were executed on the same GPU.

- 1× NVIDIA GeForce RTX 4090 D GPU (24 GB VRAM)
- 10-core/20-thread CPU
- 32 GB Memory

A.1 HYPERPARAMETERS

We report the hyperparameters used for DCFO and all baselines below. Unless otherwise specified, the same settings are used for Humanoid, Walker2d, and Hopper.

Table 3: Hyperparameters for DCFO.

Hyperparameter	**DCFO (Ours)**	**DCFO (Structure Only)**
Population size	10240	10240
Connectivity learning rate η_ρ	0.15	0.15
Structural candidate directions K_ρ	4096	4096
Structural ridge λ_ρ	10^{-4}	10^{-4}
Initial membrane time constant	10 ms	10 ms
Membrane-time-constant range $\mathcal{T}$	$[1, 20]$ ms	Fixed
Dynamical learning rate η_τ	0.10	–
Perturbation magnitude ϵ_τ	0.5 ms	–
Dynamical pairs M	512	–
Dynamical ridge λ_τ	10^{-4}	–

Table 4: Hyperparameters for EC-RSNN.

Hyperparameter	**Value**
Population size	10240
Learning rate	0.15
Exploration probability ϵ	10^{-3}
Membrane time constant	10 ms (fixed)

Table 5: Hyperparameters for Evolution Strategies baselines.

Hyperparameter	**Value**	**Target**
Population size	10240	All
Learning rate	0.15	RSNN
	0.01	GRU/LSTM
Noise standard deviation σ	0.3	RSNN
	0.02	GRU/LSTM
Weight decay	0.1	All

Table 6: Network settings.

Hyperparameter	**Value**
RSNN	
Hidden neurons H	256
Excitatory ratio	50%
Simulation timestep Δt	0.5 ms
Synaptic time constant τ_{syn}	5 ms
Output time constant τ_{out}	10 ms
GRU	
Hidden size	256
LSTM	
Hidden size	128

B Proof of the Local Descent Property

From the definition of directional credit,

$$\widehat{\mathbf{c}}_t^b = -\mathbf{V}_t^b \nabla_{x^b} \mathcal{E}_t^b(x_t^b). \tag{12}$$

Using $d_t^b = (\mathbf{V}_t^b)^\top \alpha_t^b$ and equation 8, we obtain

$$\begin{aligned} \left\langle \nabla_{x^b} \mathcal{E}_t^b(x_t^b), d_t^b \right\rangle &= -(\widehat{\mathbf{c}}_t^b)^\top \alpha_t^b \\ &= -(\alpha_t^b)^\top \left[\mathbf{V}_t^b (\mathbf{V}_t^b)^\top + \lambda_b I_{K_b} \right] \alpha_t^b \\ &= -\|d_t^b\|_2^2 - \lambda_b \|\alpha_t^b\|_2^2. \end{aligned} \tag{13}$$

Since $\lambda_b > 0$, $\mathbf{V}_t^b (\mathbf{V}_t^b)^\top + \lambda_b I_{K_b}$ is positive definite. Thus, $\widehat{\mathbf{c}}_t^b \neq \mathbf{0}$ implies $\alpha_t^b \neq \mathbf{0}$, and therefore

$$\left\langle \nabla_{x^b} \mathcal{E}_t^b(x_t^b), d_t^b \right\rangle < 0. \tag{14}$$

Hence, d_t^b is a strict local descent direction of $\mathcal{E}_t^b$ at x_t^b.

C Comparison of Membrane-Time-Constant Optimization

To isolate the effect of membrane-time-constant optimization, we compare the proposed directional credit optimization (Tau-DCFO) with an ES-based alternative (Tau-ES), while keeping the structural optimization setting unchanged.

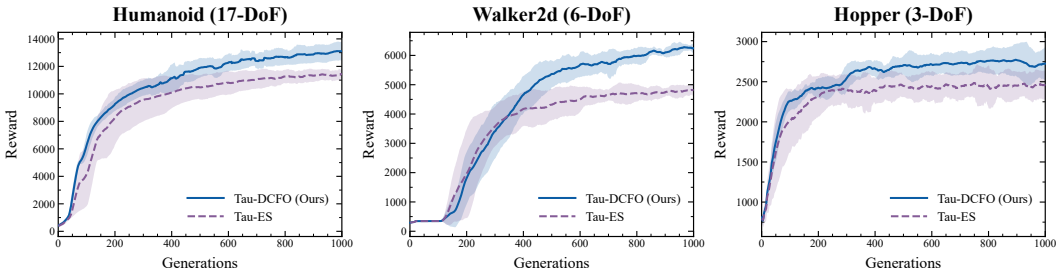

Figure 4: Comparison of Tau-DCFO and Tau-ES on Humanoid, Walker2d, and Hopper. Solid lines denote mean Reward over three seeds, and shaded regions indicate standard deviation.

As shown in Fig. 4, Tau-DCFO achieves higher Reward than Tau-ES across all three tasks, with larger improvements on Humanoid and Walker2d. This result supports the effectiveness of directional credit optimization for adapting neuron-specific membrane time constants.

D Effect of the Number of Candidate Directions

To examine the effect of directional coverage, we vary the number of structural candidate directions K_ρ while keeping the remaining training settings unchanged.

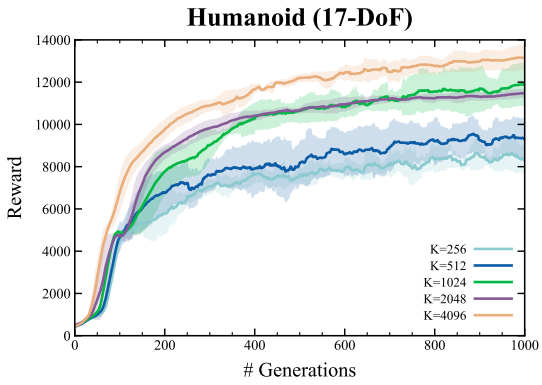

Figure 5: Effect of the number of structural candidate directions K_ρ on Humanoid. Solid lines denote mean Reward over three seeds, and shaded regions indicate standard deviation.

As shown in Fig. 5, performance generally improves with increased candidate-direction coverage, and $K_\rho = 4096$ achieves the highest final Reward. This indicates that sufficient directional coverage is important for structural credit reconstruction.

E Fixed versus Adaptive Neuronal Heterogeneity

To distinguish adaptive heterogeneity from heterogeneous initialization, we conduct a controlled comparison on the 17-DoF Humanoid task. Both settings start from the same uniformly sampled neuron-specific membrane time constants, while only DCFO updates τ during training.

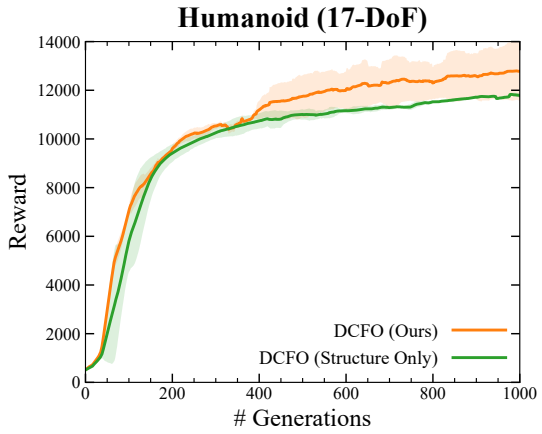

Figure 6: Comparison between fixed and adaptively optimized neuronal heterogeneity on Humanoid. Solid lines denote mean Reward over three seeds, and shaded regions indicate standard deviation.

As shown in Fig. 6, adaptive optimization of τ achieves higher final Reward than fixed heterogeneity, showing that the improvement cannot be explained by heterogeneous initialization alone.