Deep Reinforcement Learning with Spiking Neural Network for Robot Navigation and Continuous Control

Guangzhi Tang, Neelesh Kumar, Raymond Yoo

Advisor: Konstantinos Michmizos

Computational Brain Lab, Computer Science, Rutgers University

2021 Intel INRC Winter Workshop

Loihi-run robot applications at ComBra Lab

Neuromorphic SLAM

Artificial Intelligence

Automatic Driving

Disaster Rescue

Computer V

ogical Intelligence

(1D SLAM, IROS 2019)

(2D Mapping, NICE 2020)

Spiking Locomotion Control

(Spiking CPG, ICONS 2020)

(Oculomotor, BioRob 2020)*

Pros:

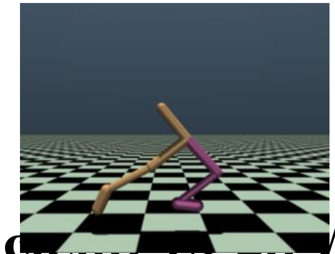
1. Fast IO and computation.
2. Consistent in repetitive and specialized tasks.

Cons:

1. **Costs a lot of energy.**
2. Lacks robustness and versatility.

Spiking Reinforcement Learning

(Navigation, IROS 2020)



(Cont. Control, CoRL 2020)

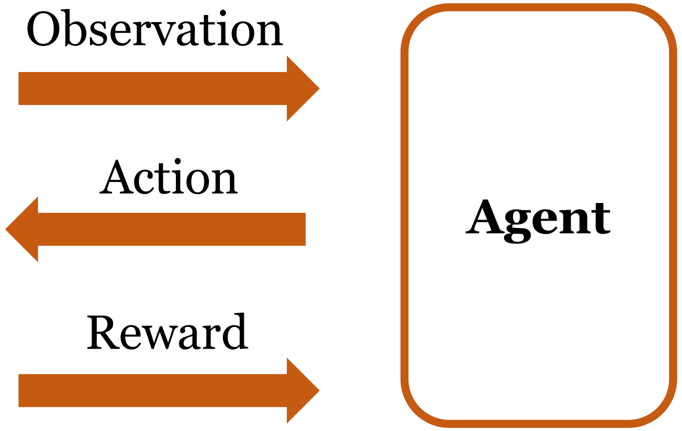
s:

1. **Energy effic Watts)**
2. Robust and versatile.

** Loihi version to be published*

Robot Planning and Control with Reinforcement Learning

Robot Interacting With Environment



Maximize Cumulative Reward

Observation: LiDAR, Point Clouds, IMU, Tactile ...

Action: Wheel Velocity, Joint Force, Decision ...

Reward: Goal Reaching, Collision, Moving Speed, ...

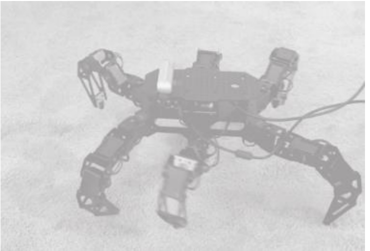
Loihi-run robot applications at ComBra Lab

Neuromorphic SLAM

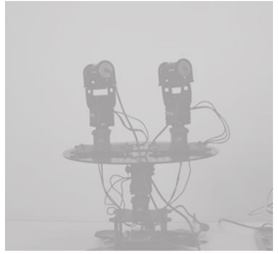
(1D SLAM, IROS 2019)

(2D Mapping, NICE 2020)

Spiking Locomotion Control



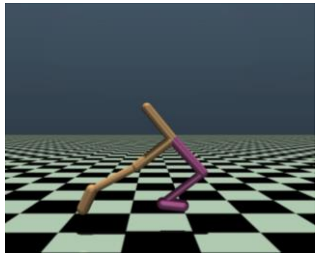
(Spiking CPG, ICONS 2020)



(Oculomotor, BioRob 2020)*

Spiking Reinforcement Learning

(Navigation, IROS 2020)



(Cont. Control, CoRL 2020)

** Loihi version to be published*

Reinforcement co-Learning for Mapless Navigation

Hybrid Training with SDDPG

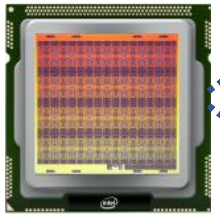
Deep Critic Network ⇄ Spiking Actor Network

- DNN and SNN are trained jointly using gradient descent
- Hybrid training allows DNN and SNN to overcome each other's limitations through a shared representation learning

Energy-Efficient Mapless Navigation

Observations

In



Intel's Loihi

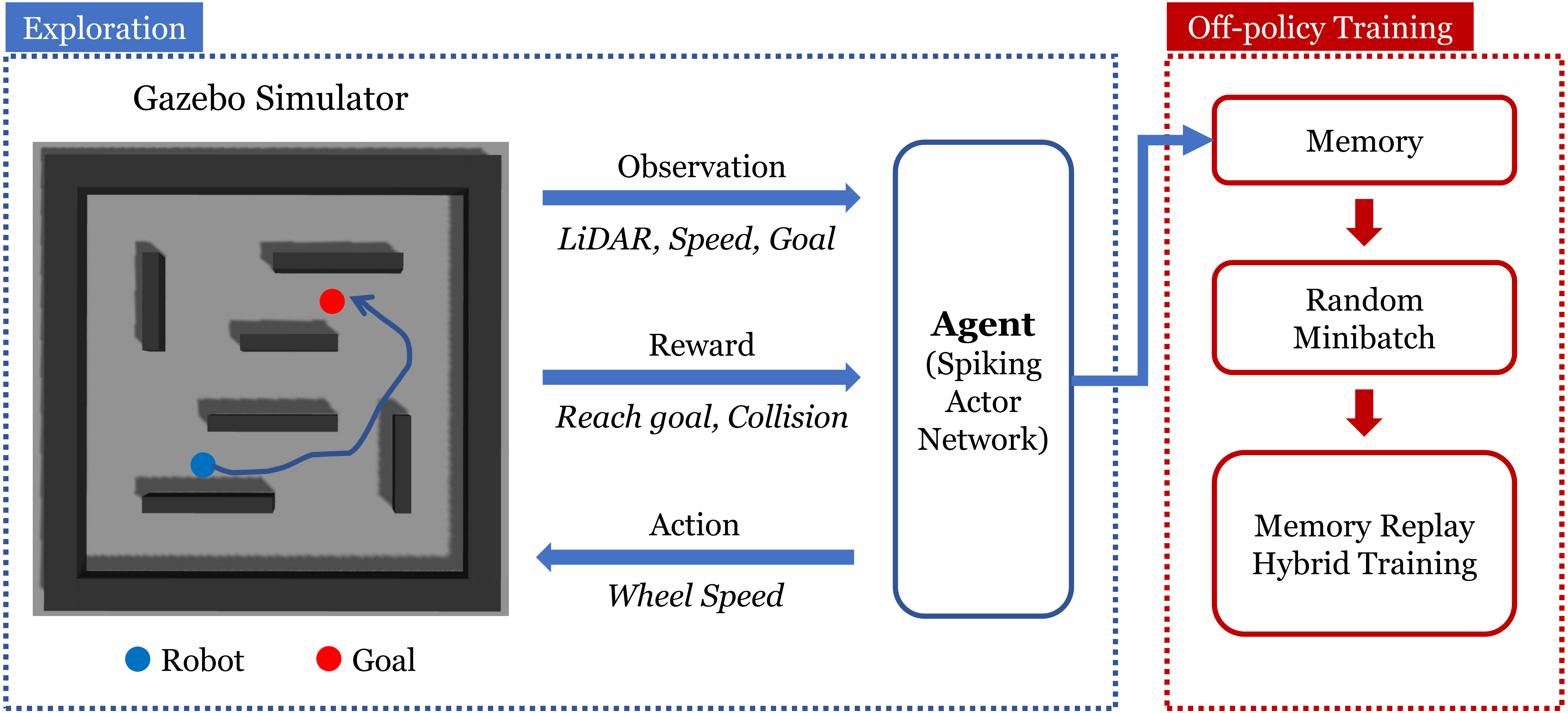
Out

Differential Wheel Speed

Control robot toward goals with obstacle avoidance

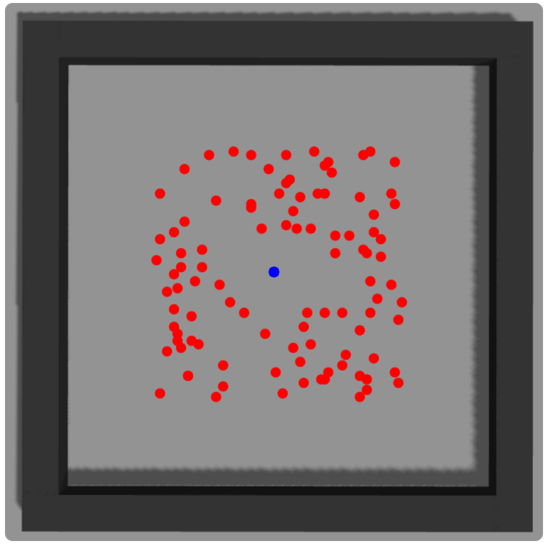
Guangzhi Tang, et al. "Reinforcement co-Learning of Deep and Spiking Neural Networks for Energy-Efficient Mapless Navigation with Neuromorphic Hardware." IROS 2020

Overview of Mapless Navigation Training

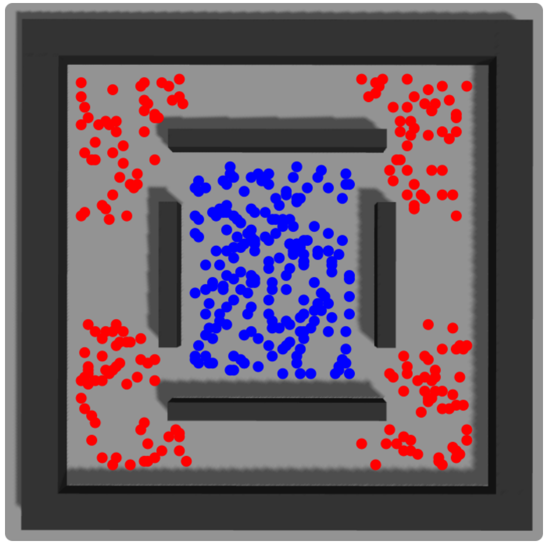


Curriculum Learning: Train from easy to hard

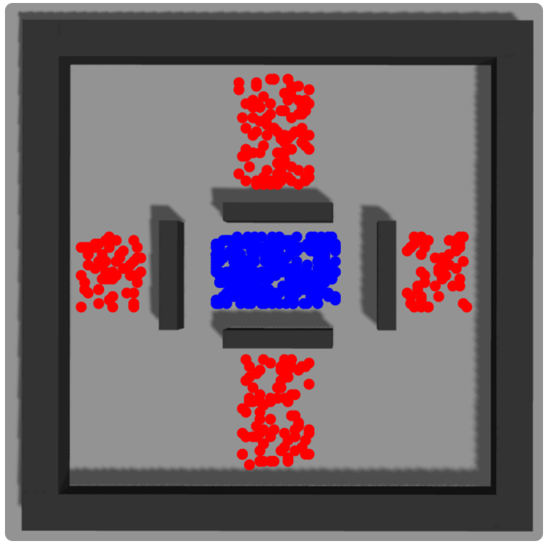
(Easy) Train Sequentially with Random Start-Goal Pairs (Hard)



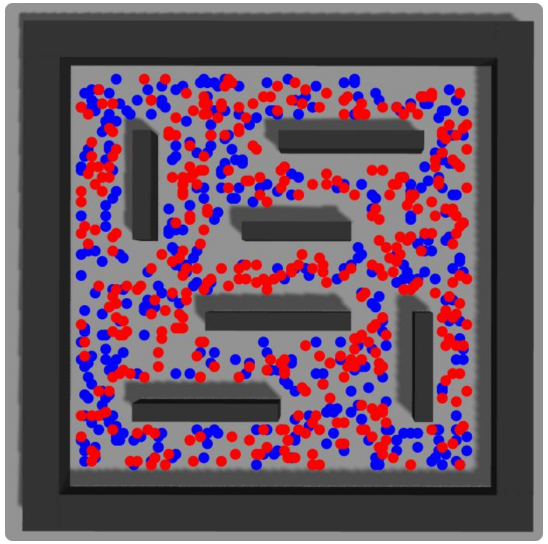
Free Path



Partially Block Path

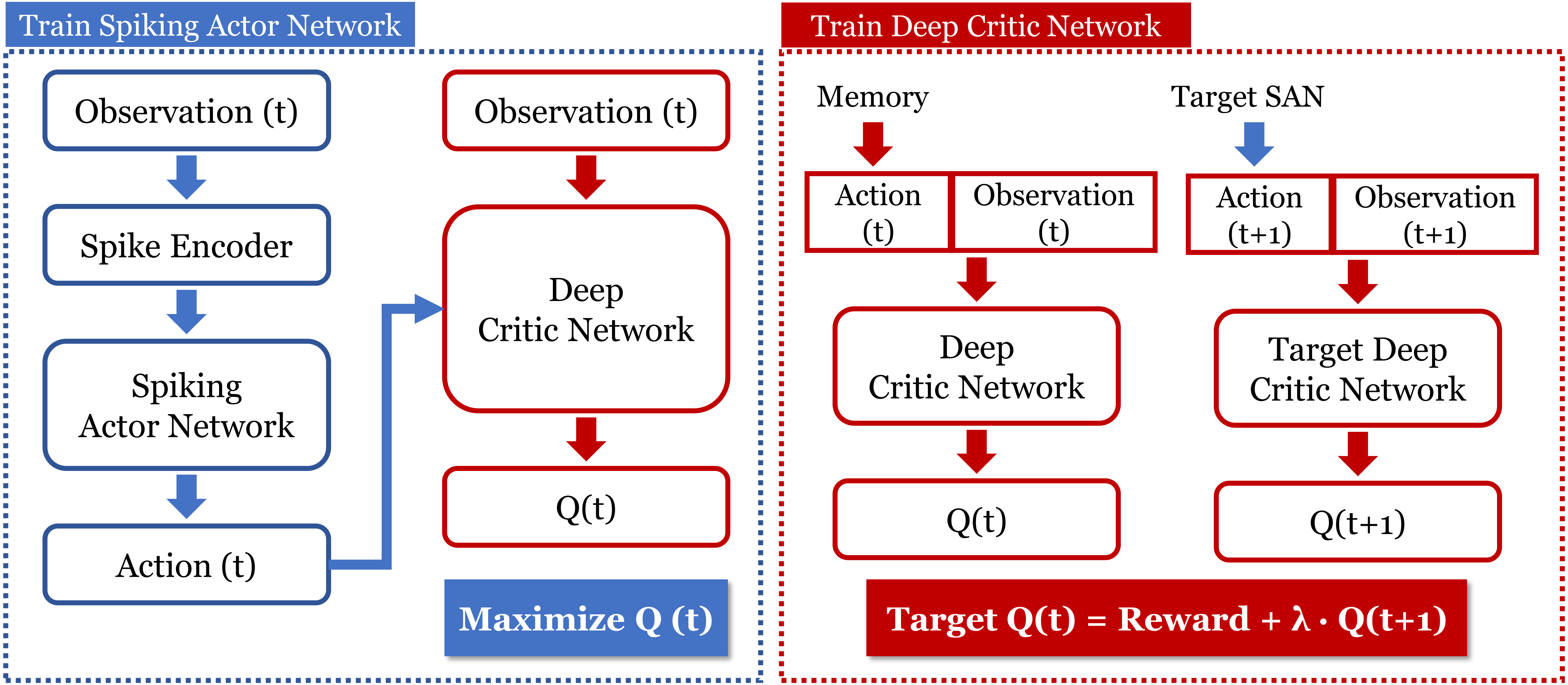


Fully Block Path

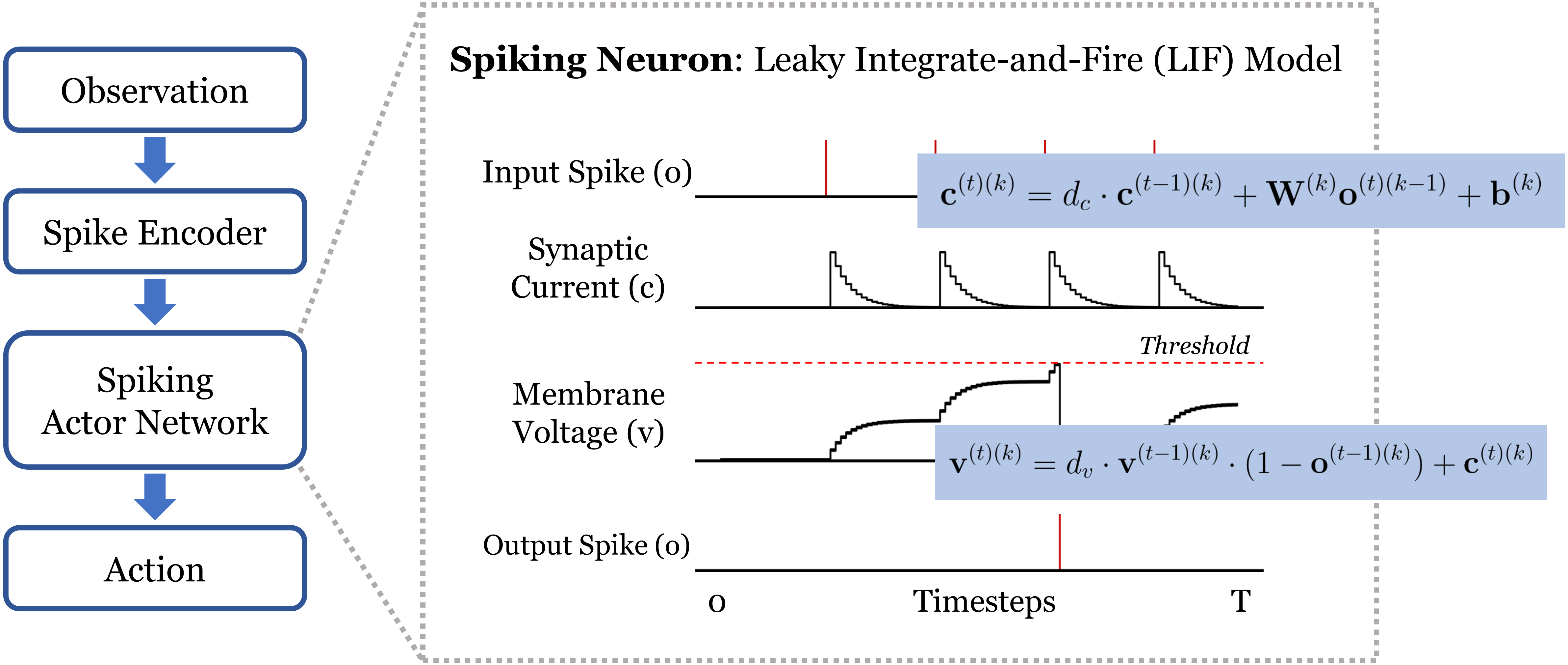


Everything Combined

SDDPG: Spiking Deep Deterministic Policy Gradient

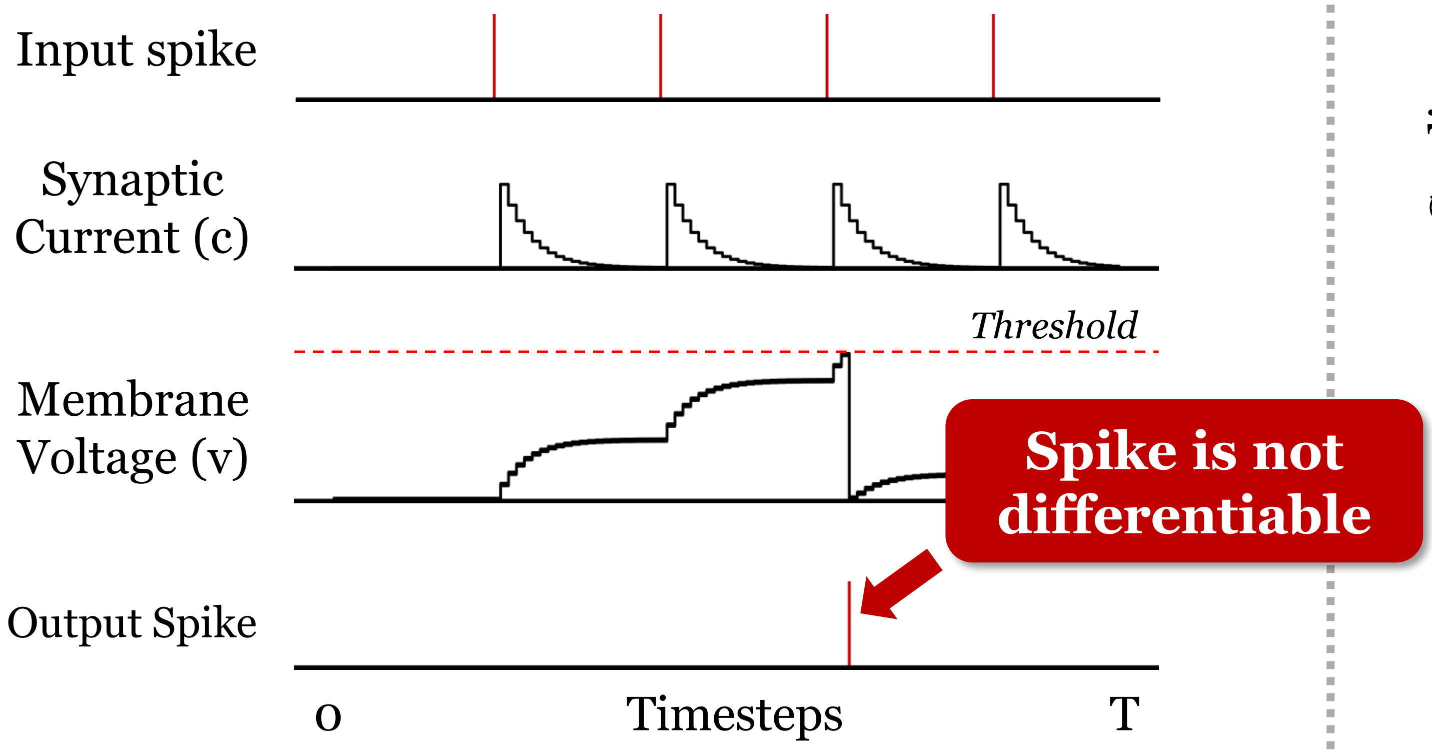


Spiking Actor Network: Forward Propagation

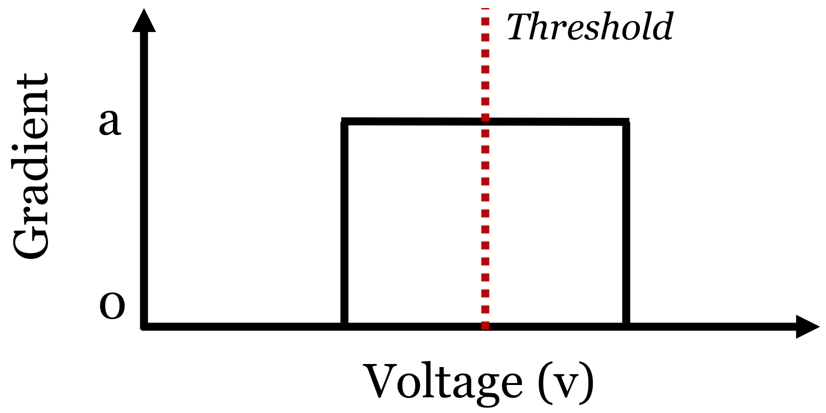


Spiking Actor Network: Backward Propagation

Spiking Neuron: Leaky Integrate-and-Fire (LIF) Model



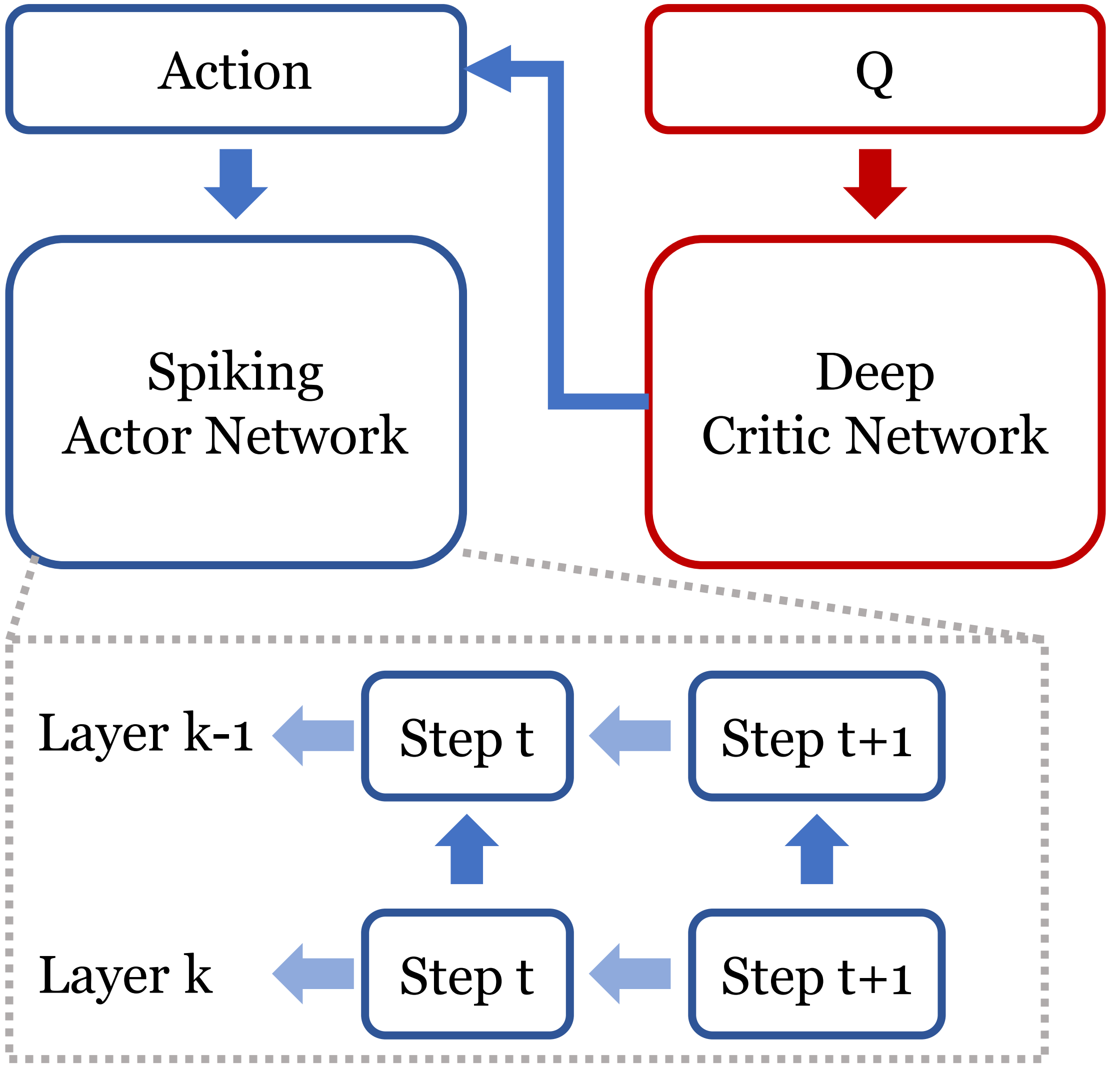
Approximate the gradient of a spike



Rectangular function

$z\left(\mathbf{v}^{(t)(k)}\right)$

Spiking Actor Network: Backward Propagation



Maximize action value Q with loss function:

$$L = -Q$$

Gradient propagates through critic network:

$$\nabla_{\mathbf{Action}} L = \mathbf{W}_c^{(n+1)'} \cdot \nabla_{a^{(n+1)}} L$$

Spatial and temporal gradients of spiking neurons:

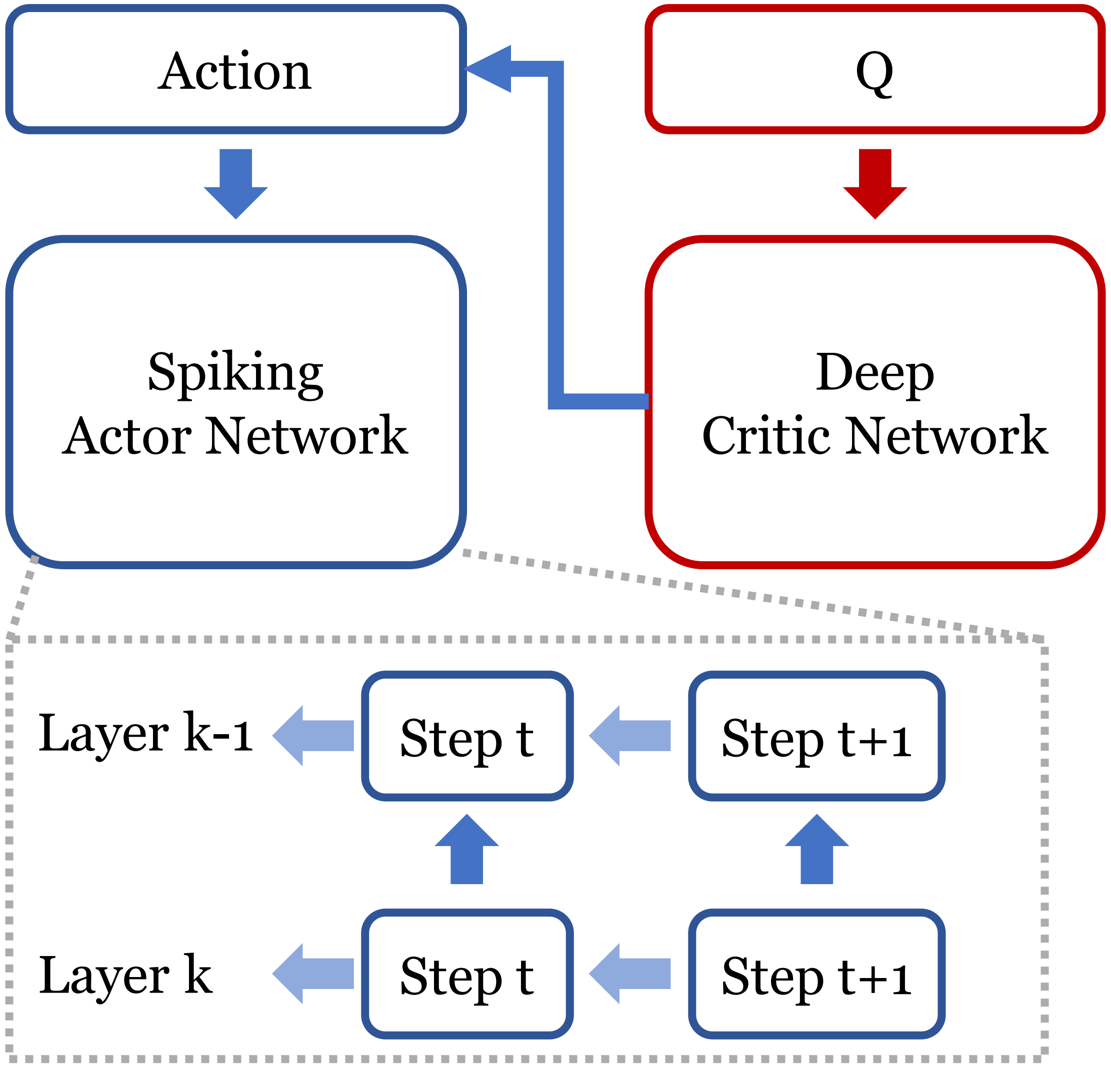
$$\nabla_{\mathbf{v}^{(t)(k)}} L = \overbrace{z(\mathbf{v}^{(t)(k)}) \cdot \nabla_{\mathbf{o}^{(t)(k)}} L}^{Spatial} + \overbrace{d_v(1 - \mathbf{o}^{(t)(k)}) \cdot \nabla_{\mathbf{v}^{(t+1)(k)}} L}^{Temporal}$$

$$\nabla_{\mathbf{c}^{(t)(k)}} L = \overbrace{\nabla_{\mathbf{v}^{(t+1)(k)}} L}^{Spatial} + \overbrace{d_c \nabla_{\mathbf{c}^{(t+1)(k)}} L}^{Temporal}$$

$$\nabla_{\mathbf{o}^{(t)(k-1)}} L = \mathbf{W}^{(k)'} \cdot \nabla_{\mathbf{c}^{(t)(k)}} L$$

Propagate gradient to presynaptic layer

Spiking Actor Network: Backward Propagation



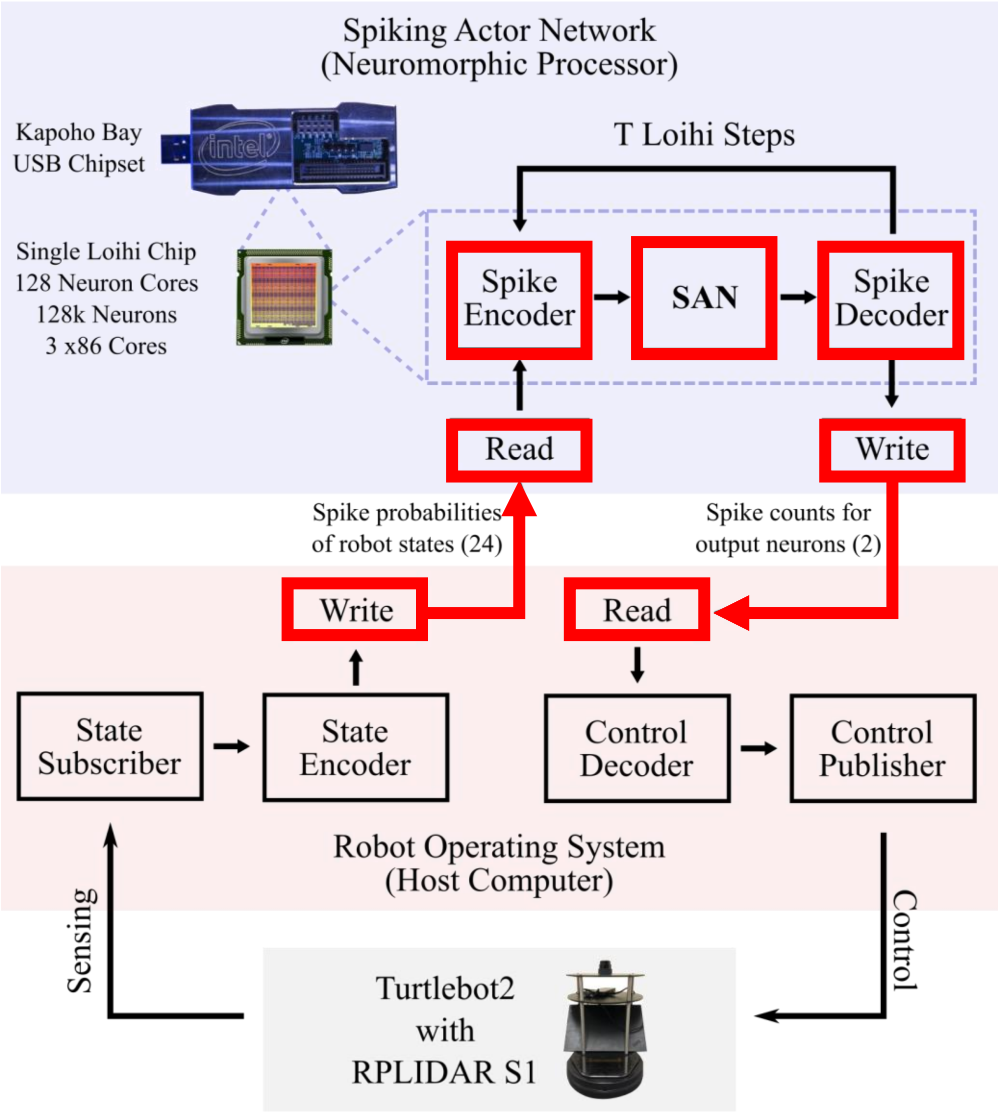
Combining gradients from all timesteps and update weights and biases:

$$\nabla_{\mathbf{W}^{(k)}} L = \sum_{t=1}^{T} \mathbf{o}^{(t)(k-1)} \cdot \nabla_{\mathbf{c}^{(t)(k)}} L$$

$$\nabla_{\mathbf{b}^{(k)}} L = \sum_{t=1}^{T} \nabla_{\mathbf{c}^{(t)(k)}} L$$

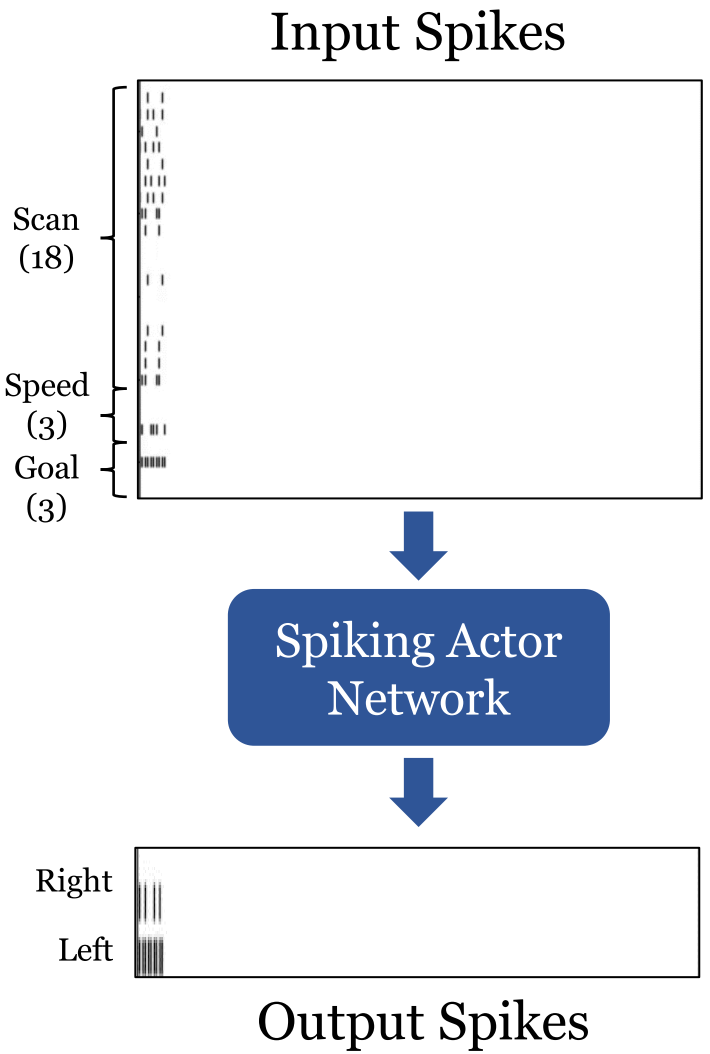
Update network parameters every T timesteps

Realization on Loihi Neuromorphic Processor

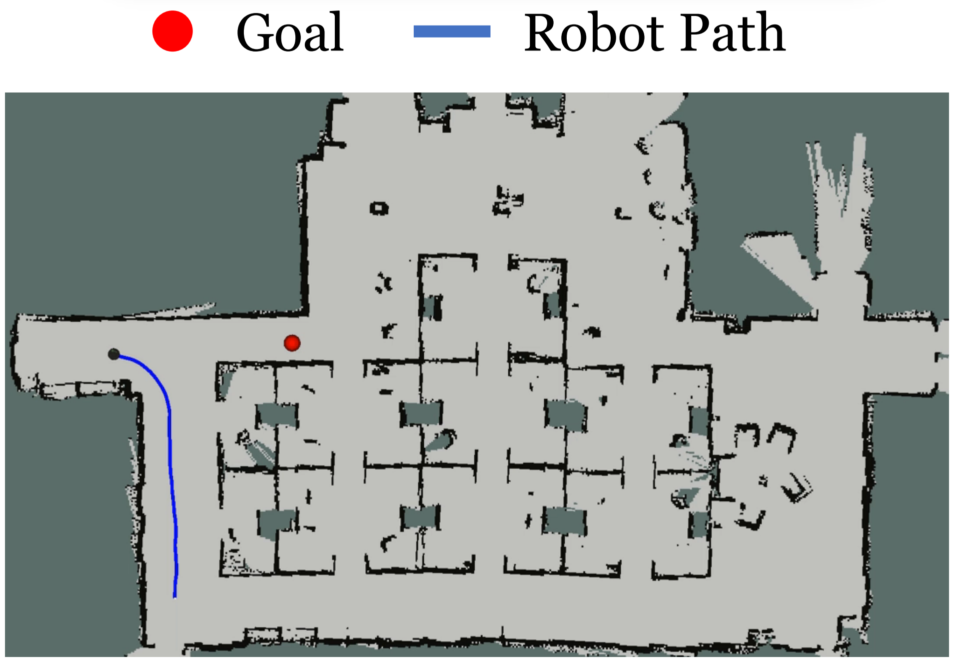


- Realize the trained SAN onto Loihi with low precision weights using layer-wise rescaling
- ROS communicates with Loihi using data channels
- Deploy encoder and decoder on Loihi's low-frequency x86 cores to reduce data transfer load
- ROS-Loihi interaction framework controls the mobile robot in real-time

Spiking Neural Network for Autonomous Navigation



Real-world evaluation for SNN on Loihi



Speed x2

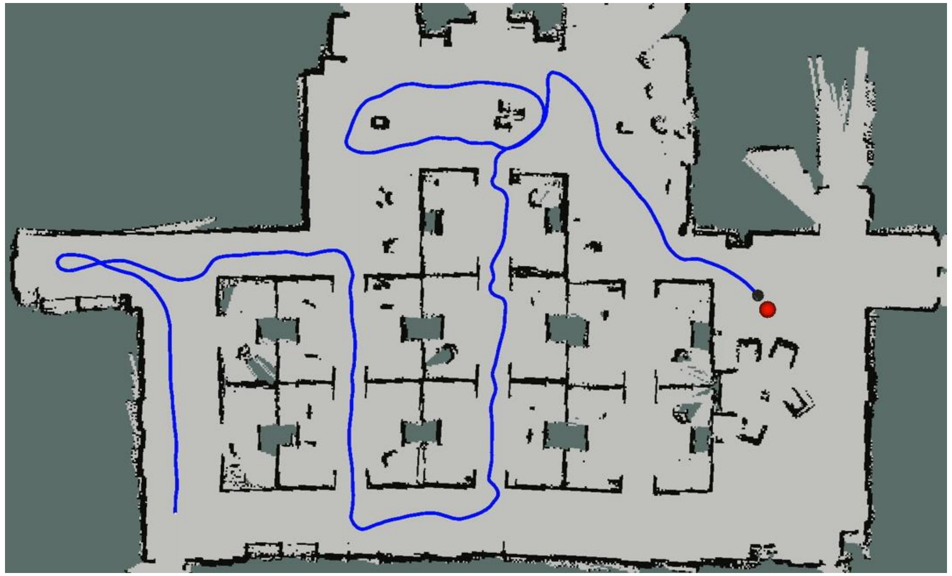
Spiking Neural Network for Autonomous Navigation

Deep Network on Jetson TX2

SNN on Loihi

SNN is **75** times more energy efficient than Deep Network

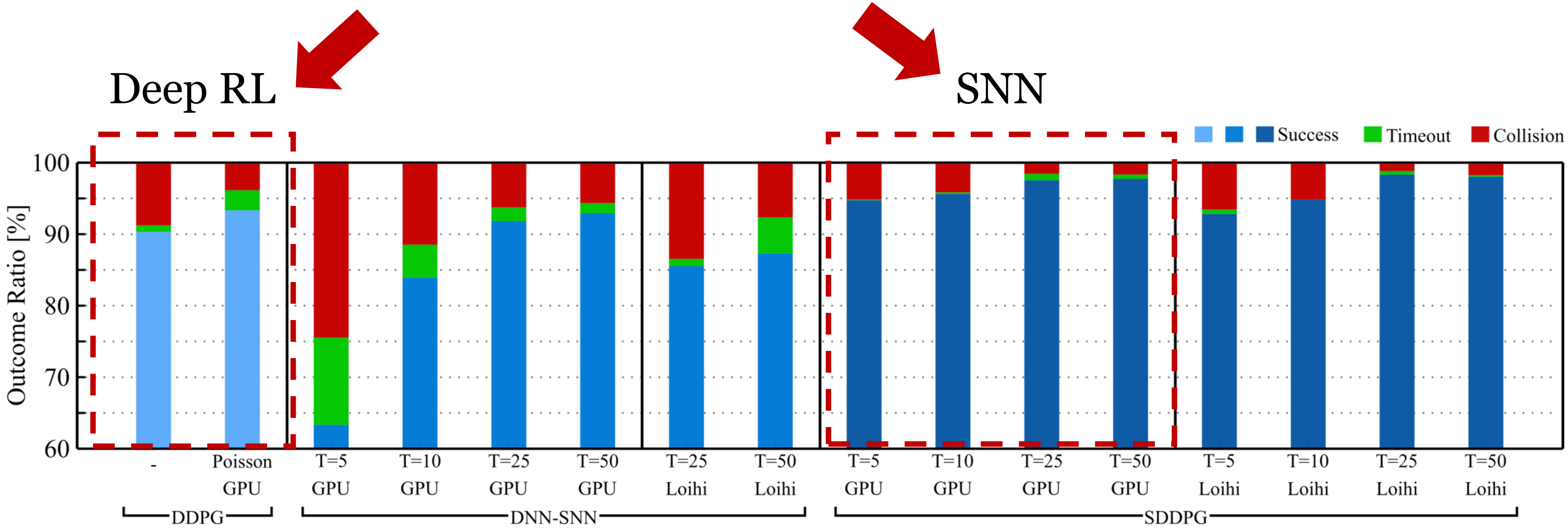
● Goal — Robot Path



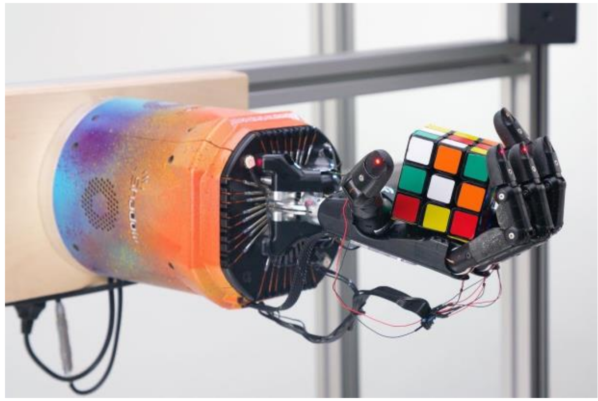
Speed x4

Spiking Neural Network for Autonomous Navigation

SNN shows higher successful rate navigating in complex environment

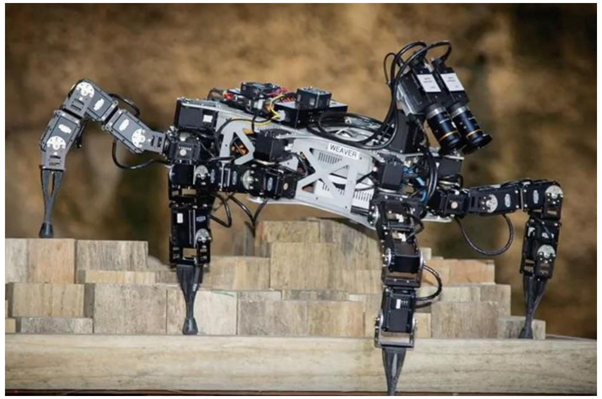


High-dimensional Continuous Control



OpenAI Robot Hand

Google Robot Arm Farm

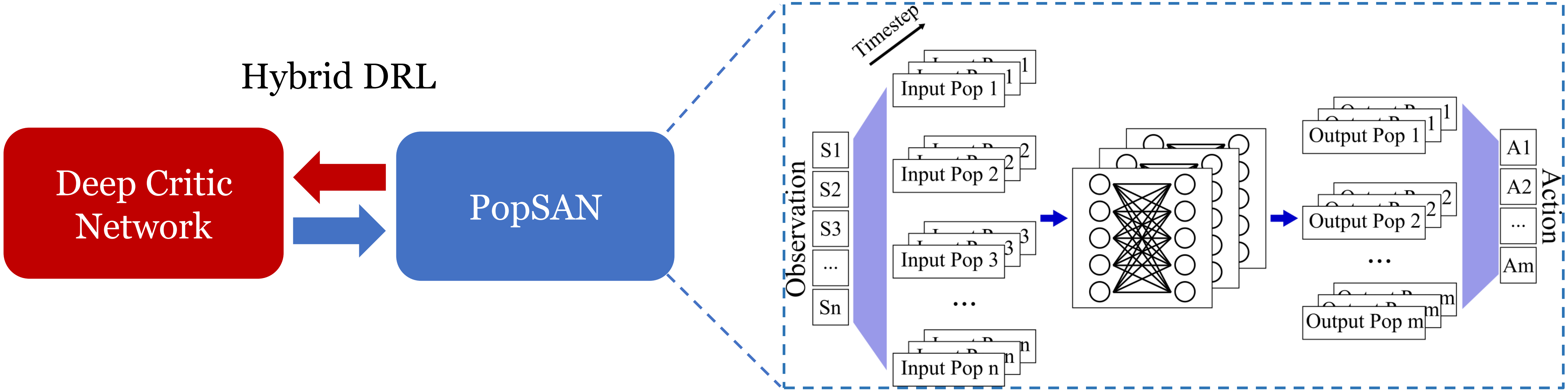


CSIRO Weaver Hexapod

Boston Dynamic Spot Robot

- Robot systems for complex applications often have high-dimensional observation and action space
- Optimality of the control policy highly depends on the encoding precision of the continuous observation and action
- Encoding precision of individual spiking neuron is limited due to event-based computation
- Especially problematic when a small inference timestep is used for better energy efficiency

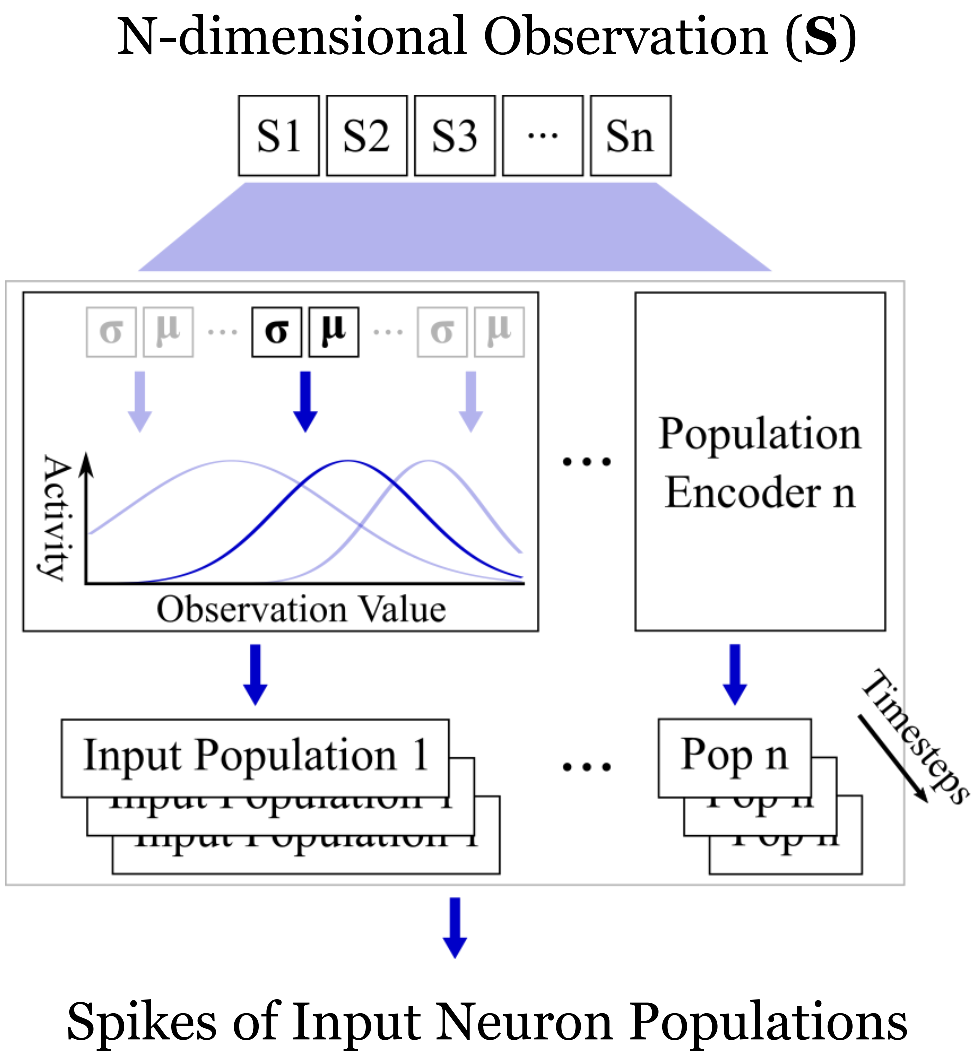
Population-coded Spiking Actor Network (PopSAN)



- Encodes each dimension of the observation and action spaces in individual neuron populations with learnable receptive fields
- Supports a wide spectrum of DRL algorithms (DDPG, TD3, SAC, and PPO) to learn energy-efficient solutions for continuous control problems

Guangzhi Tang, et al. "Deep Reinforcement Learning with Population-coded Spiking Neural Network for Continuous Control." CoRL 2020

Input Neuron Populations in PopSAN

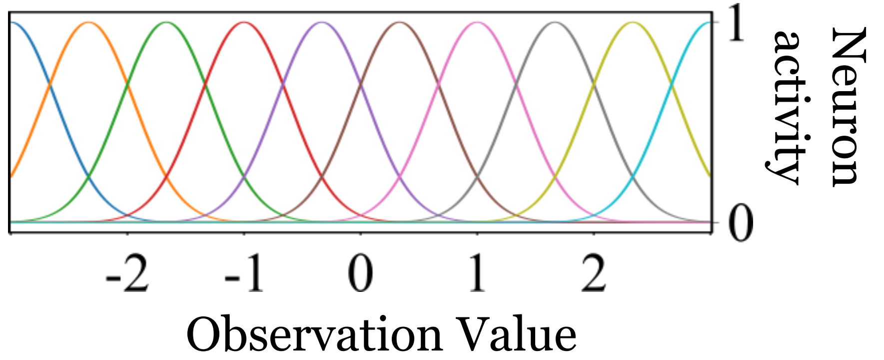


Each dimension of **S** is encoded by a population of neurons with neuron activity defined by:

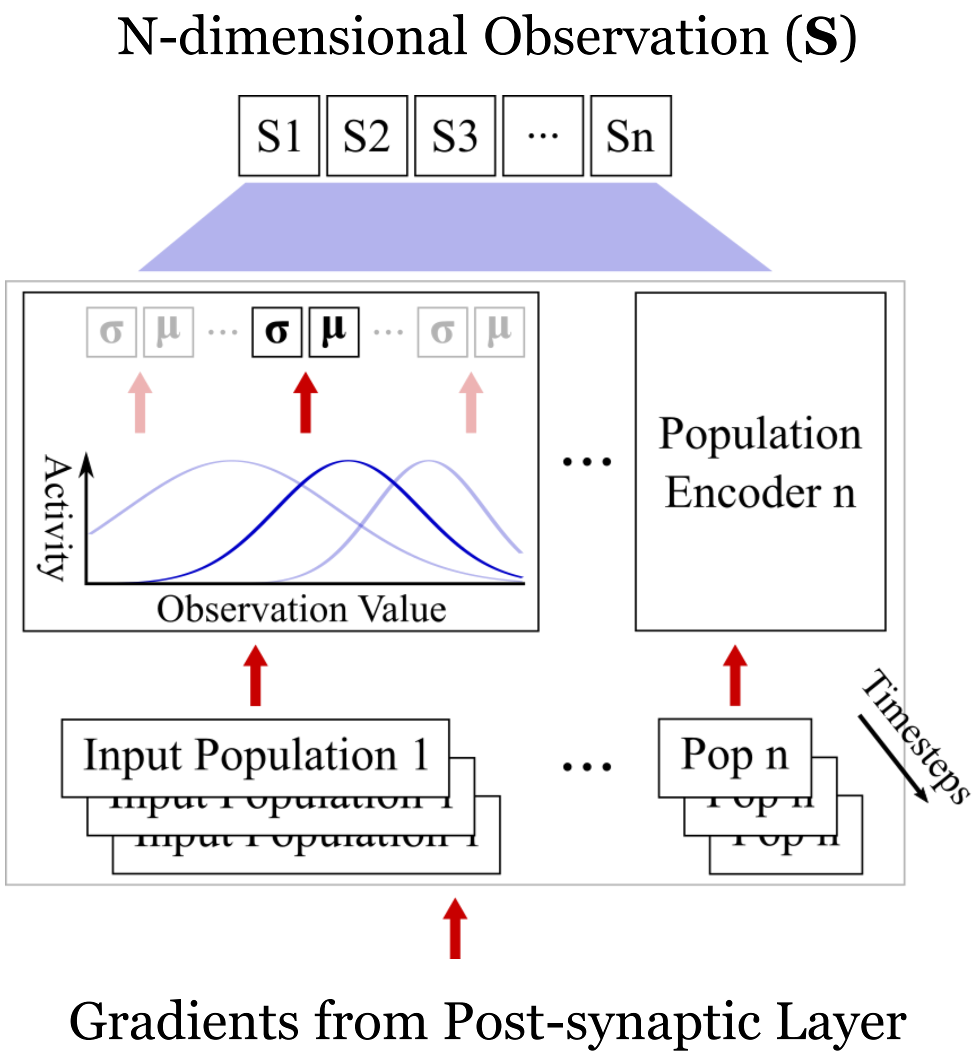
$$A_E = exp(-1/2 \cdot ((S_i - \mu)/\sigma)^2)$$

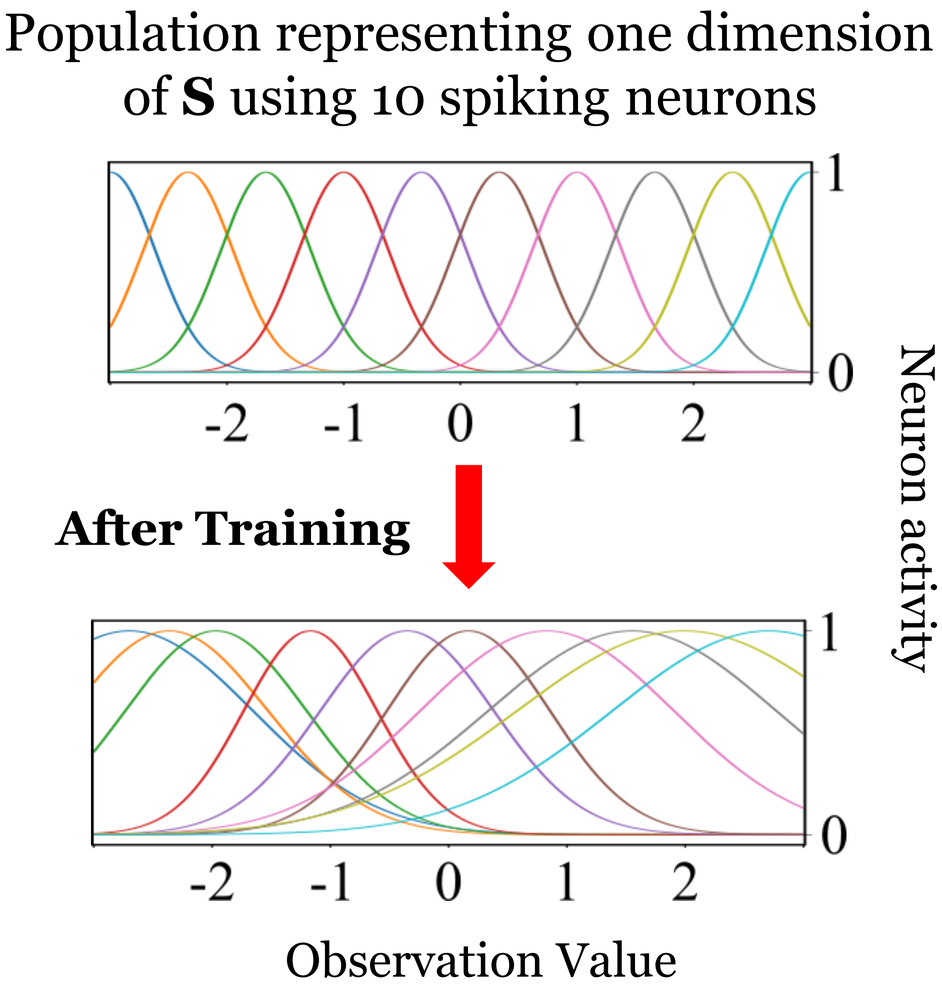
Neuron receptive field is defined by (**μ**, **σ**), which are trainable

Population representing one dimension of **S** using 10 spiking neurons

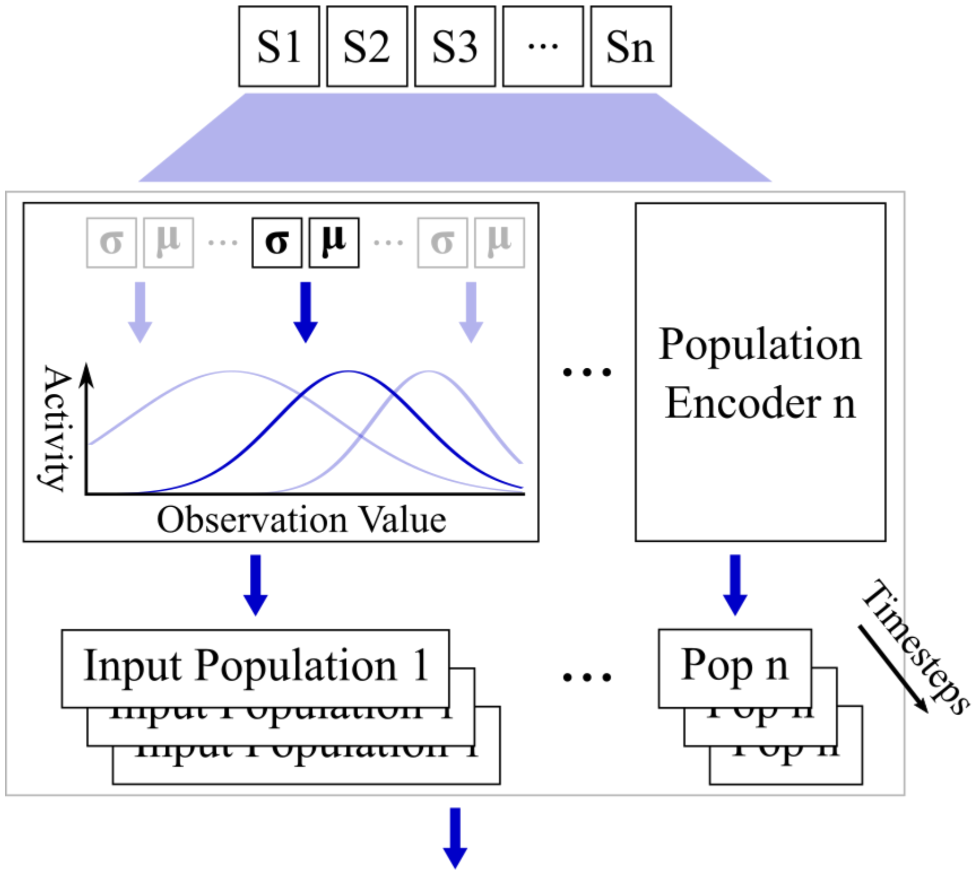


Input Neuron Populations in PopSAN

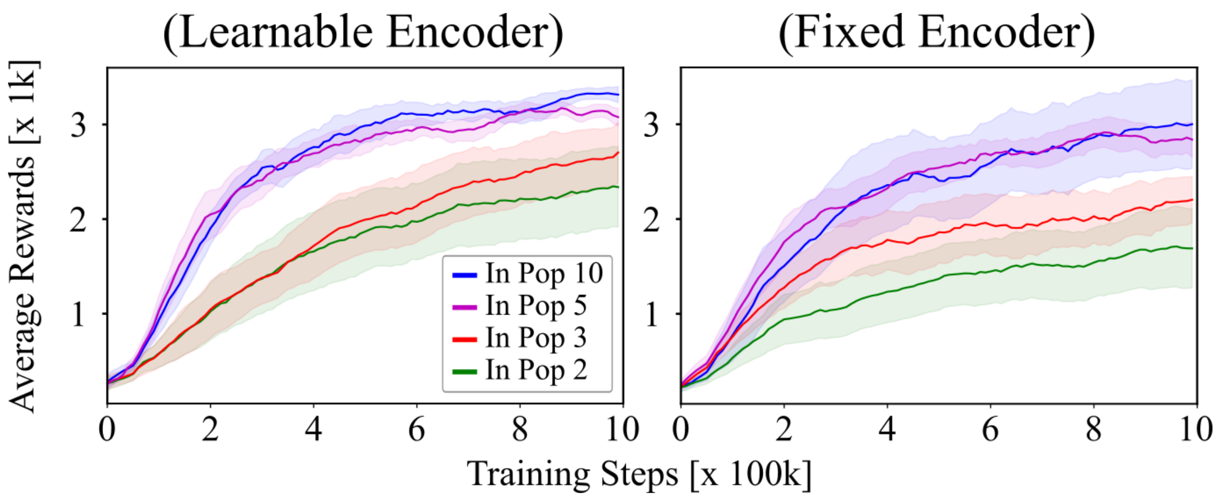




Input Neuron Populations in PopSAN

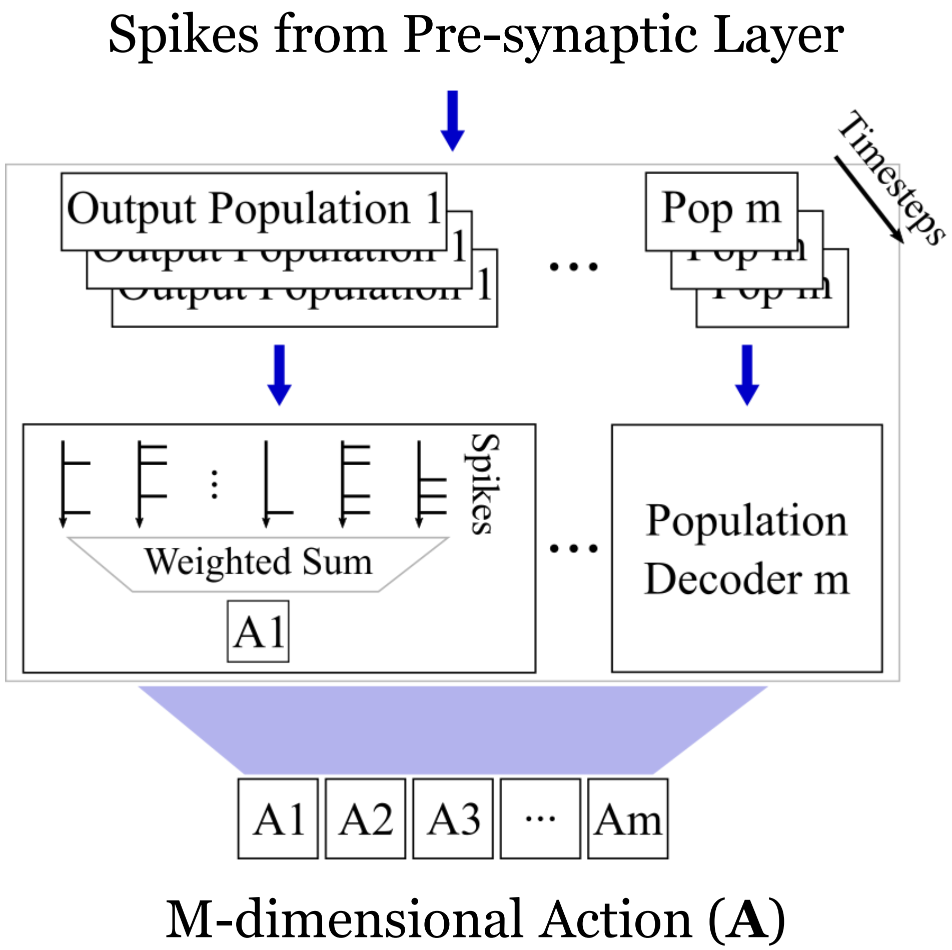


PopSAN trained using TD3 for Hopper-v3



- Learnable encoder performs better than fixed encoder
- Decrease of population size hurts the performance of PopSAN

Output Neuron Populations in PopSAN



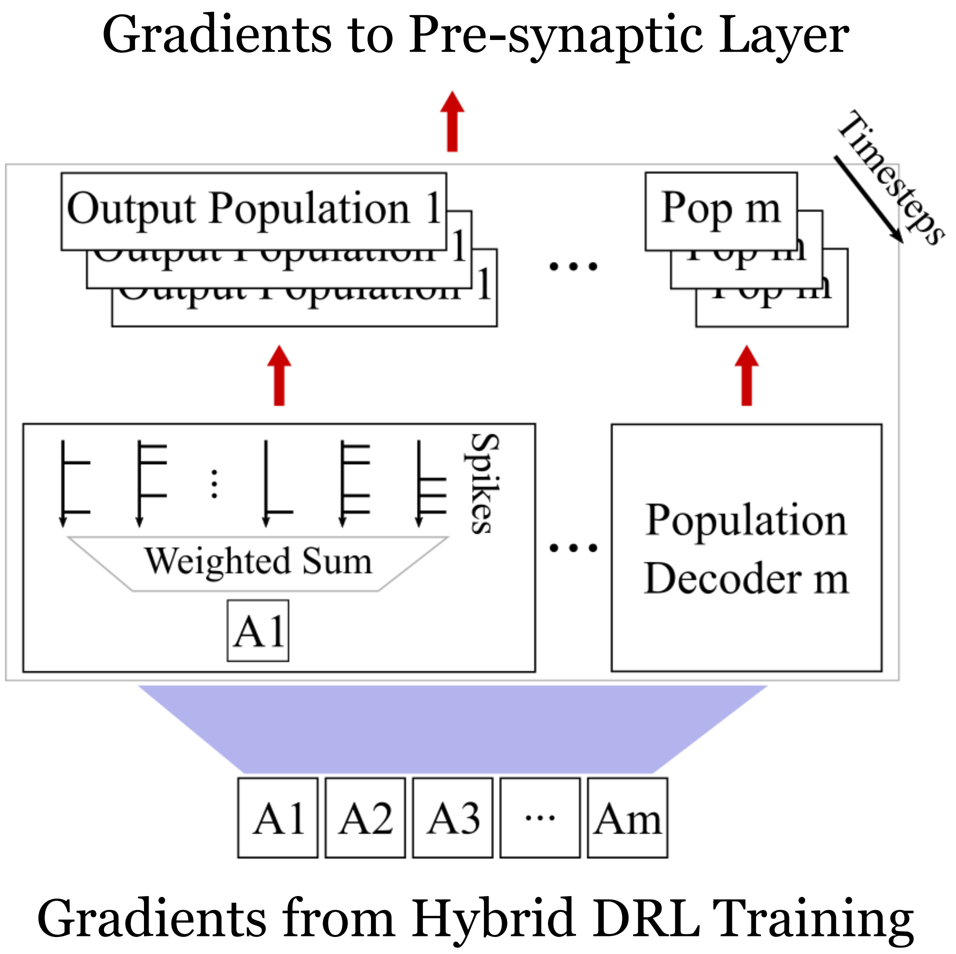
Output populations are fully connected to the last hidden layer of PopSAN.

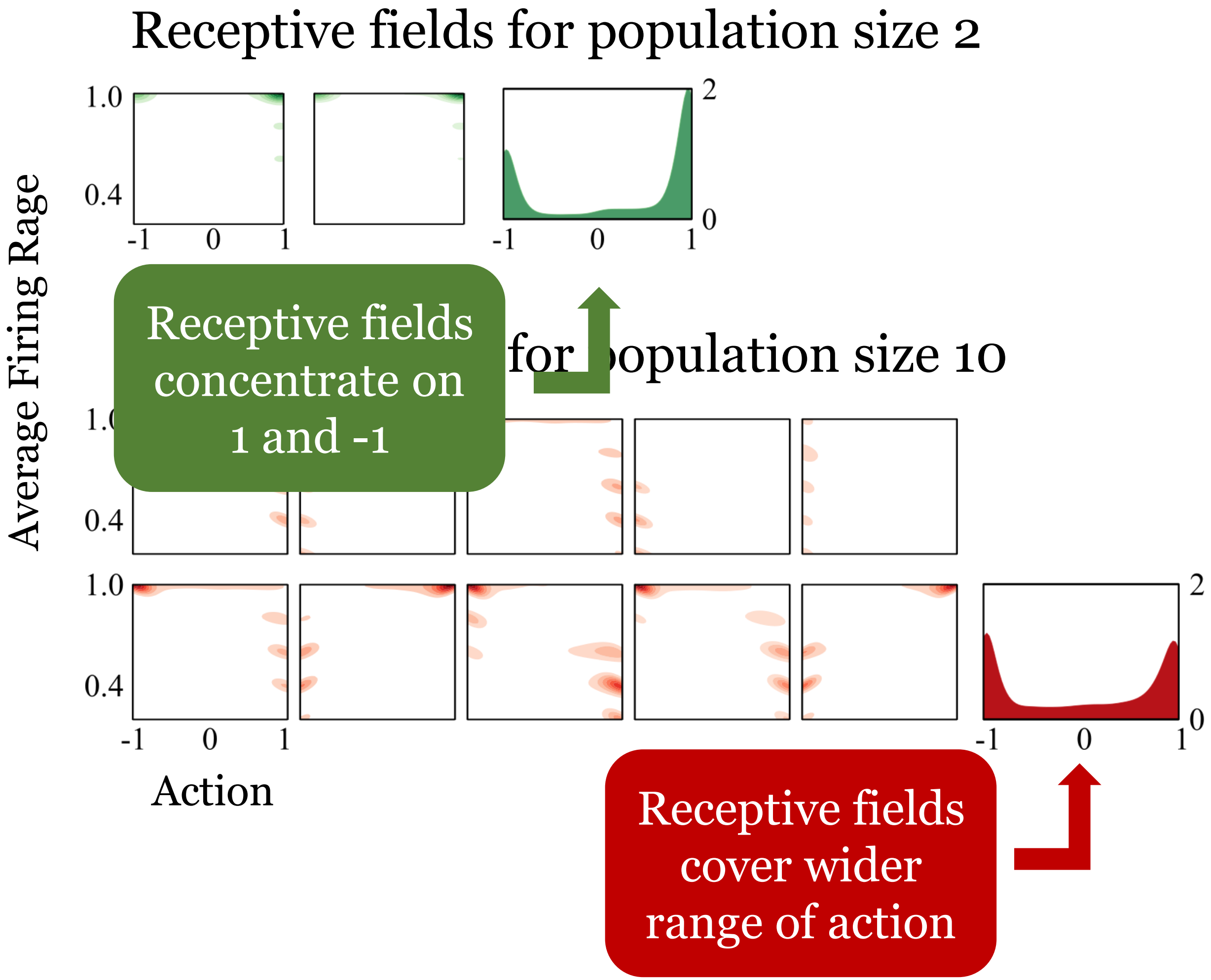
Each dimension of A is decoded by the weighted sum of neuron activities in the population:

$$A_i = \frac{1}{T}\sum_{t=1}^{T} W_i O_i(t)$$

Neuron receptive field is defined by W, which are trainable

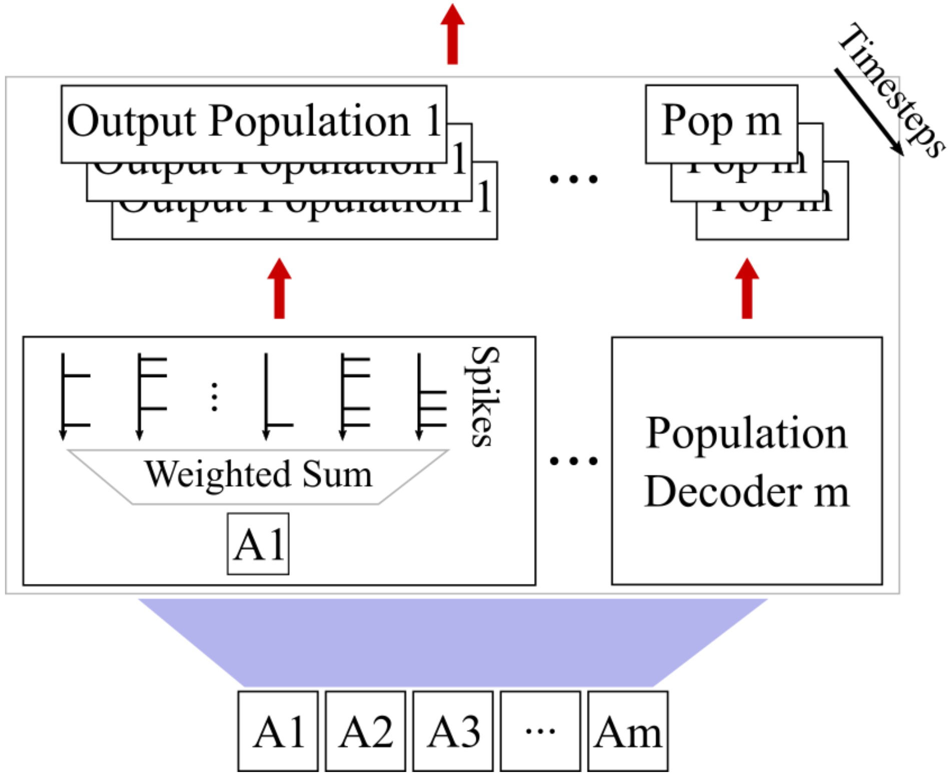
Output Neuron Populations in PopSAN



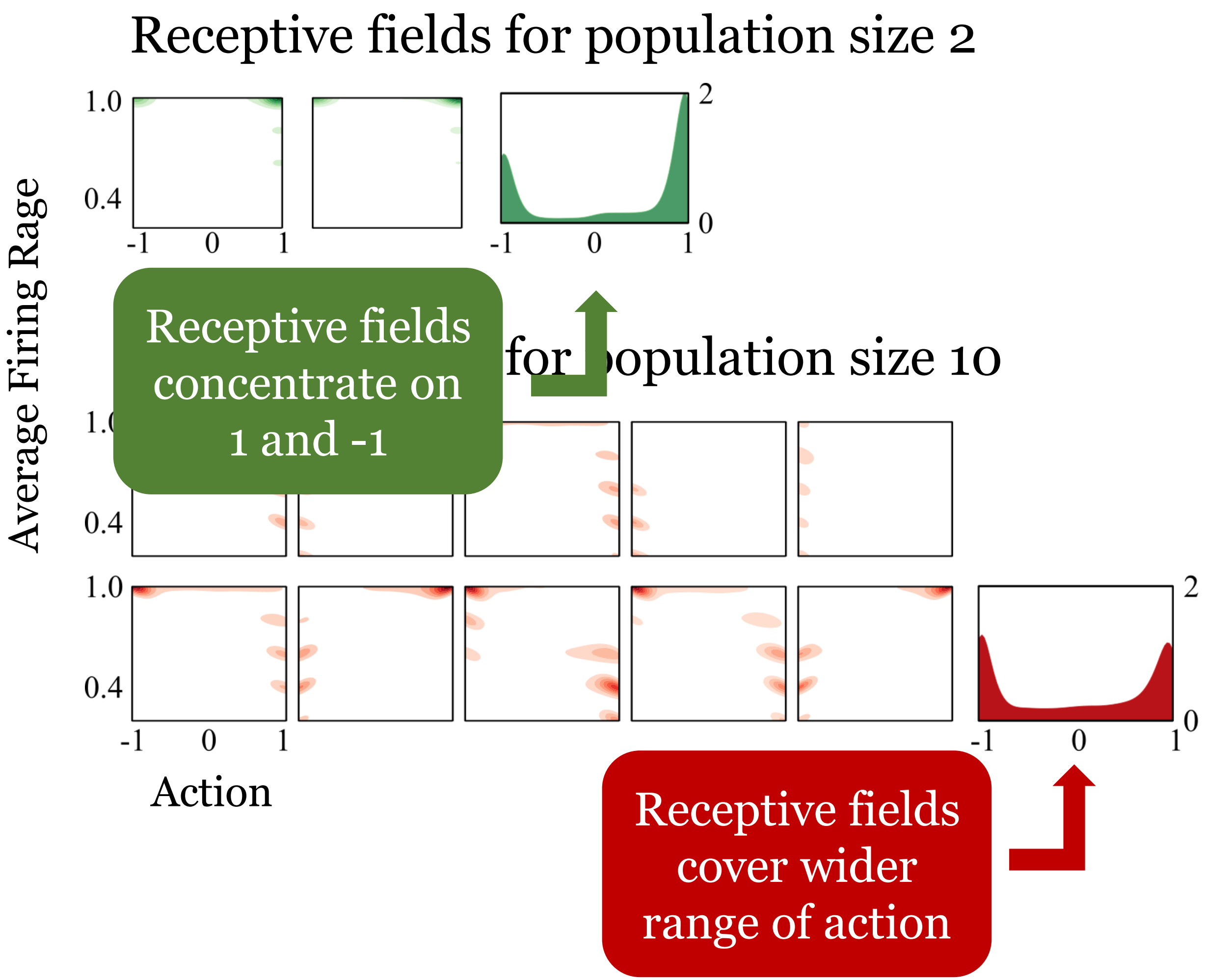


Output Neuron Populations in PopSAN

Gradients to Pre-synaptic Layer

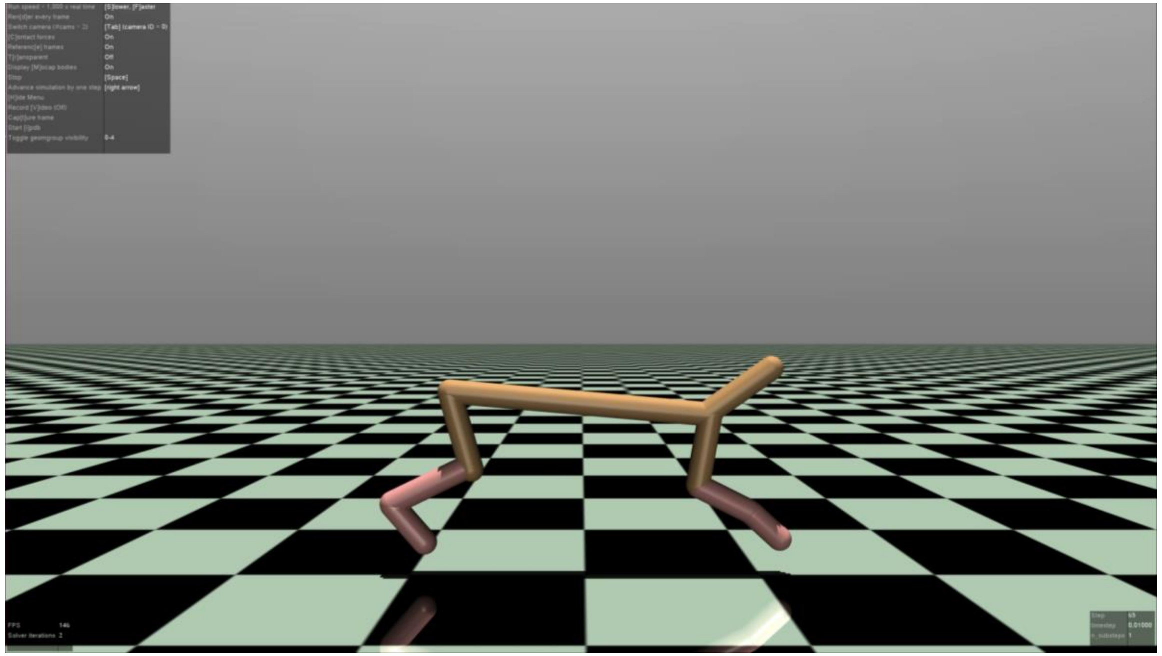


Gradients from Hybrid DRL Training

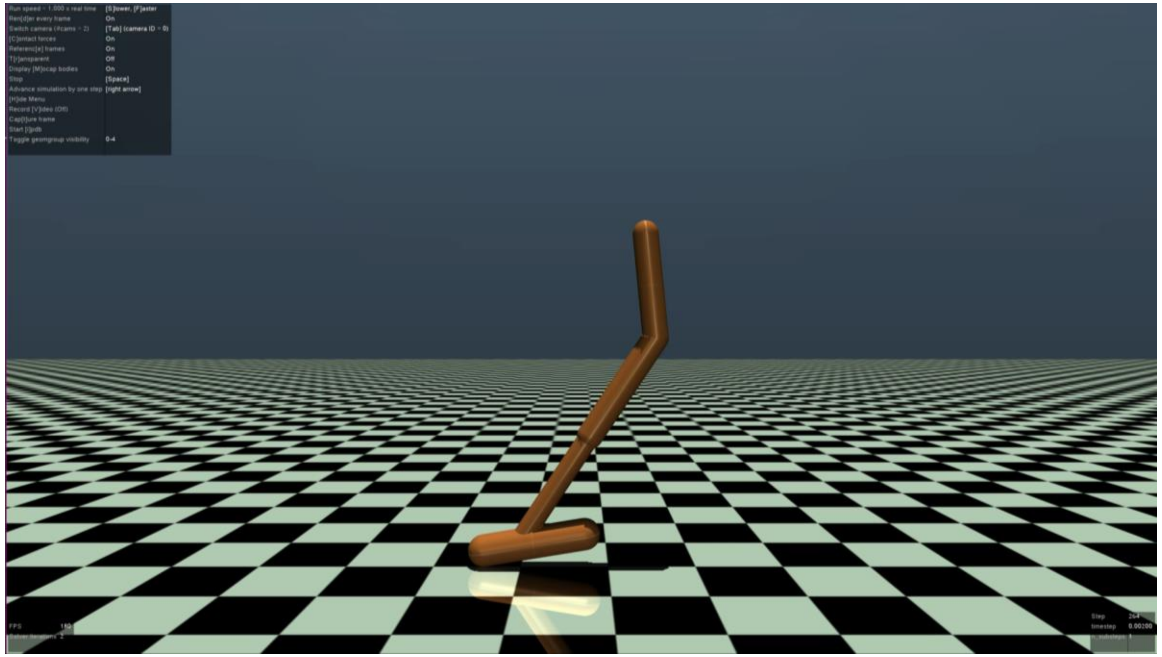


Real-time Loihi Control with PopSAN

HalfCheetah



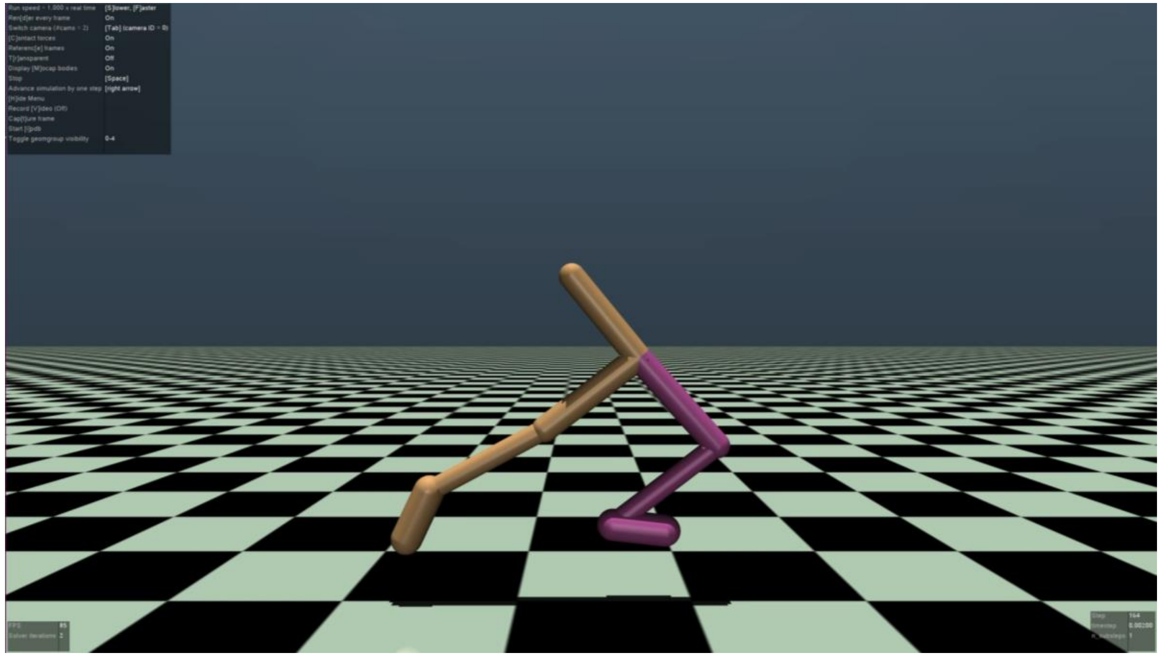
Hopper



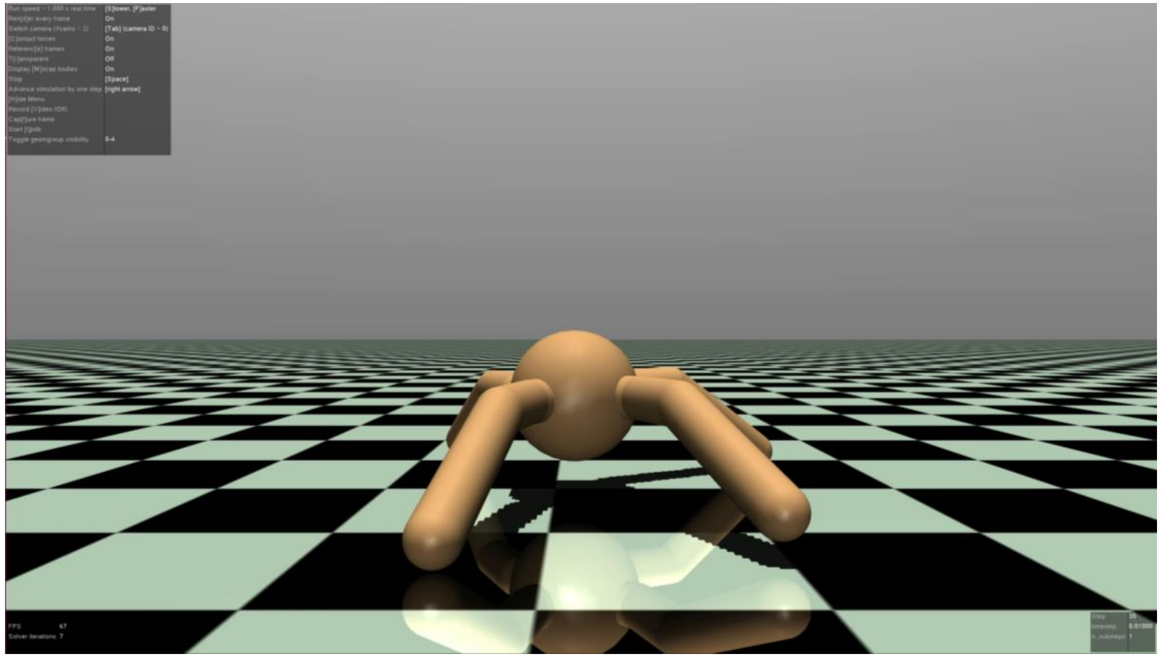
(Trained using TD3 algorithm)

Real-time Loihi Control with PopSAN

Walker2d

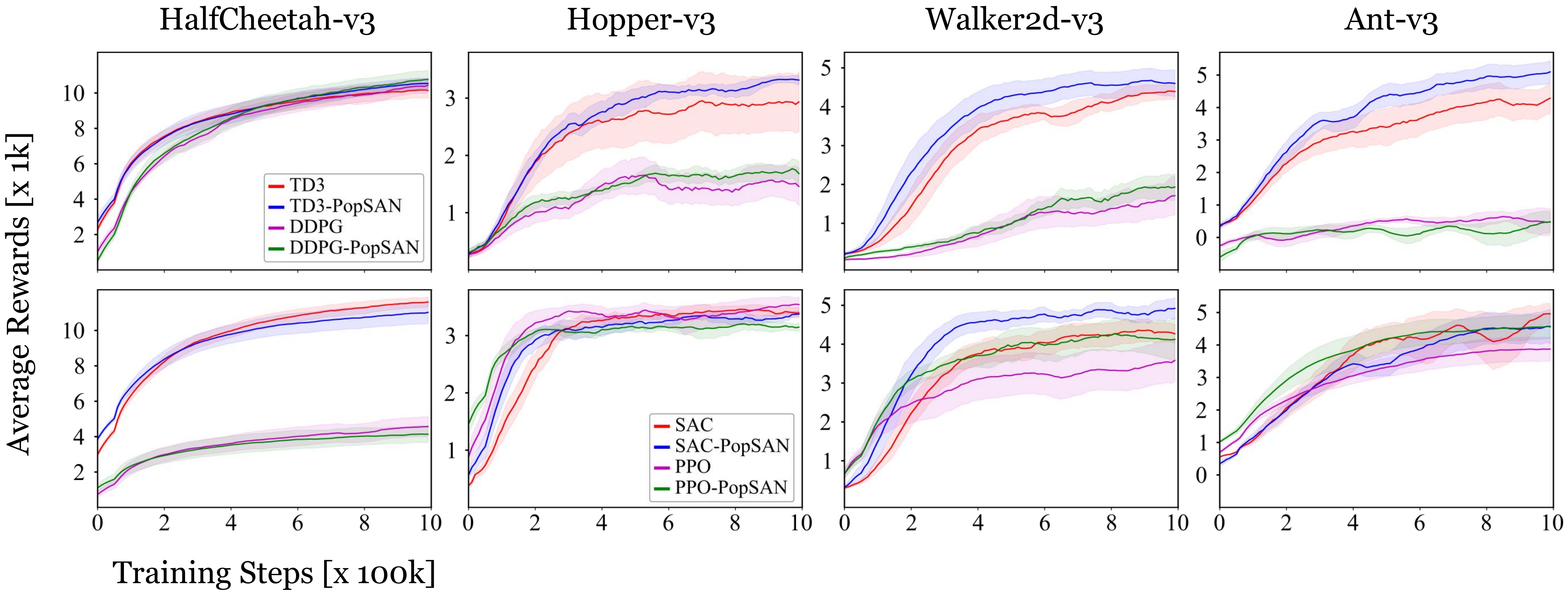


Ant



(Trained using TD3 algorithm)

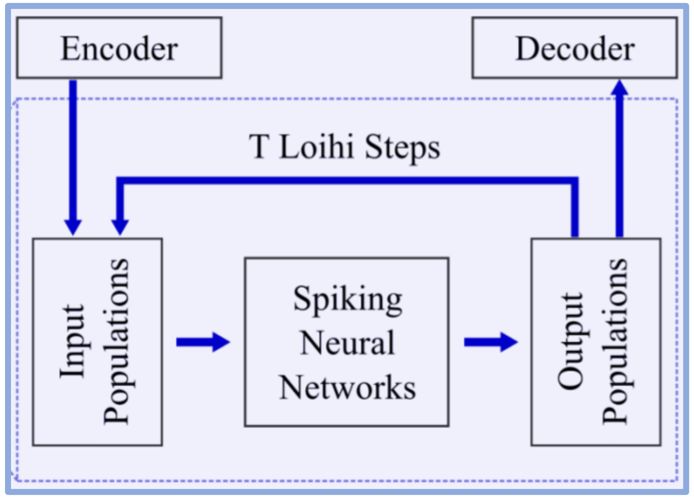
PopSAN trained using on-policy and off-policy DRL



PopSAN achieved **the same level of performance** as the Deep Actor Networks across all tasks, while **140 times more energy efficient** during inference.

Challenges for Robot Control and DRL with Loihi

Inference Control Loop on Loihi



Limited inference speed with Loihi when compared with other processors for the same control task

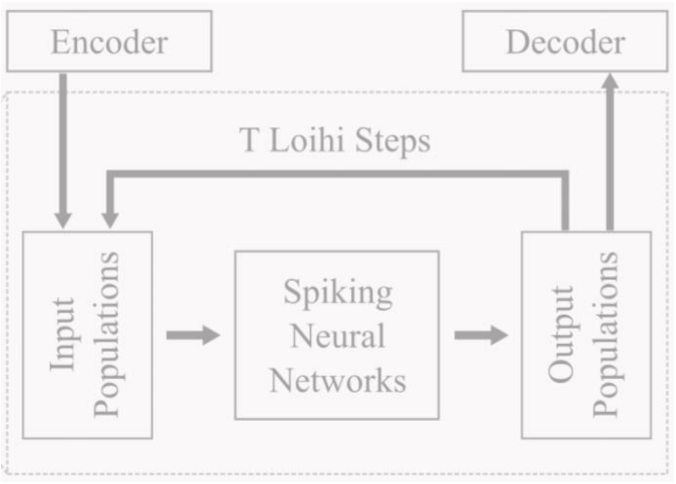
- Frame-based inference requires multiple Loihi steps to produce one control command
- Input/Output bandwidth becomes a bottleneck of Loihi for high-dimensional observation and action

Method	Device	Idle (W)	Dynamic (W)	Inf/s	[illegible]
DNN	CPU	15.51	58.93	7450	7909.86
DNN	GPU	24.68	46.04	3782	12174.46
DNN	TX2(N)	1.24	1.76	750	2346.71
DNN	TX2(Q)	[illegible]	[illegible]	[illegible]	[illegible]
PopSAN	Loihi	1.084	0.003	226	**1198**

Power performance and inference speed across hardware

Challenges for Robot Control and DRL with Loihi

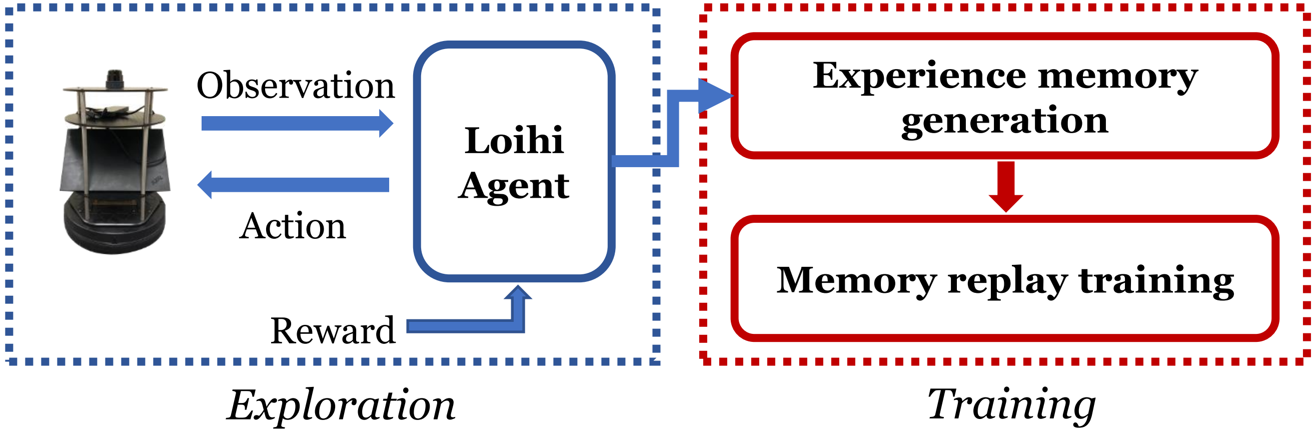
Inference Control Loop on Loihi



Limited inference speed with Loihi when compared with other processors for the same control task

- Frame-based inference requires multiple Loihi steps to produce one control command
- Input/Output bandwidth becomes a bottleneck of Loihi for high-dimensional observation and action

Real-world Robotic Reinforcement Learning



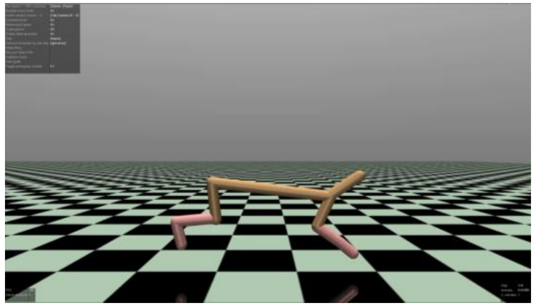
Extend our approach to a continuous and autonomous robotic learning system on Loihi for real-world applications

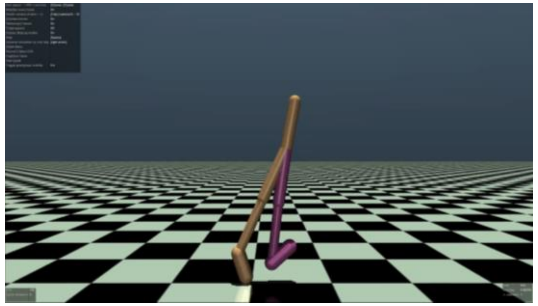
- Experience generation replacing experience storage
- Minibatch training and high-precision gradient encoding

In Conclusion:

- Towards energy-efficient and versatile robotic perception and control solutions applicable to Loihi-controlled mobile robots.

- General solution for training spiking neural network in complex real-world reinforcement learning applications.





This research is supported by Intel's Neuromorphic Research Community (INRC) Global Research Award