ICML
International Conference
On Machine Learning

Session 4, Topic: Zero-order and Black-box Optimization

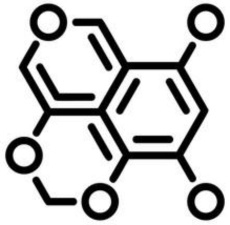
# EARL-BO: Reinforcement Learning for Multi-Step Lookahead, High-Dimensional Bayesian Optimization

ICML 2025

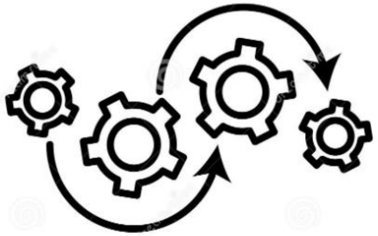
Authors: Mujin Cheon, Dong Yeun Koh, Jay H. Lee, and Calvin Tsay

KAIST IMPERIAL USC

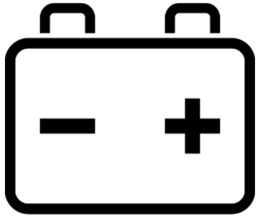
# Black-box optimization



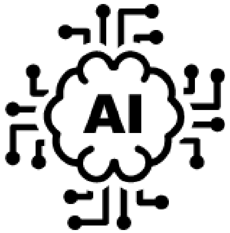
Material discovery



Process optimization

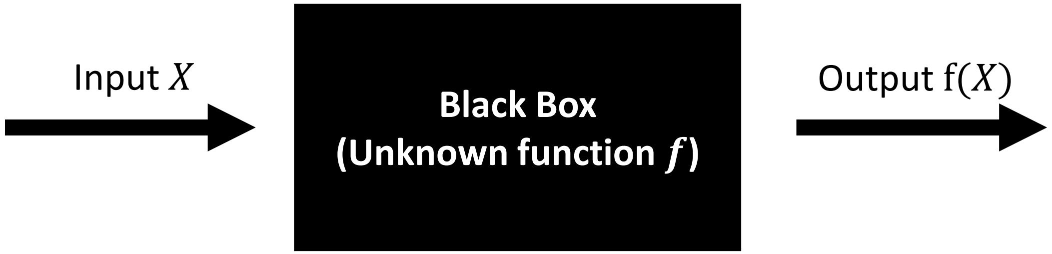


Design optimization



Training hyperparameters

⋮



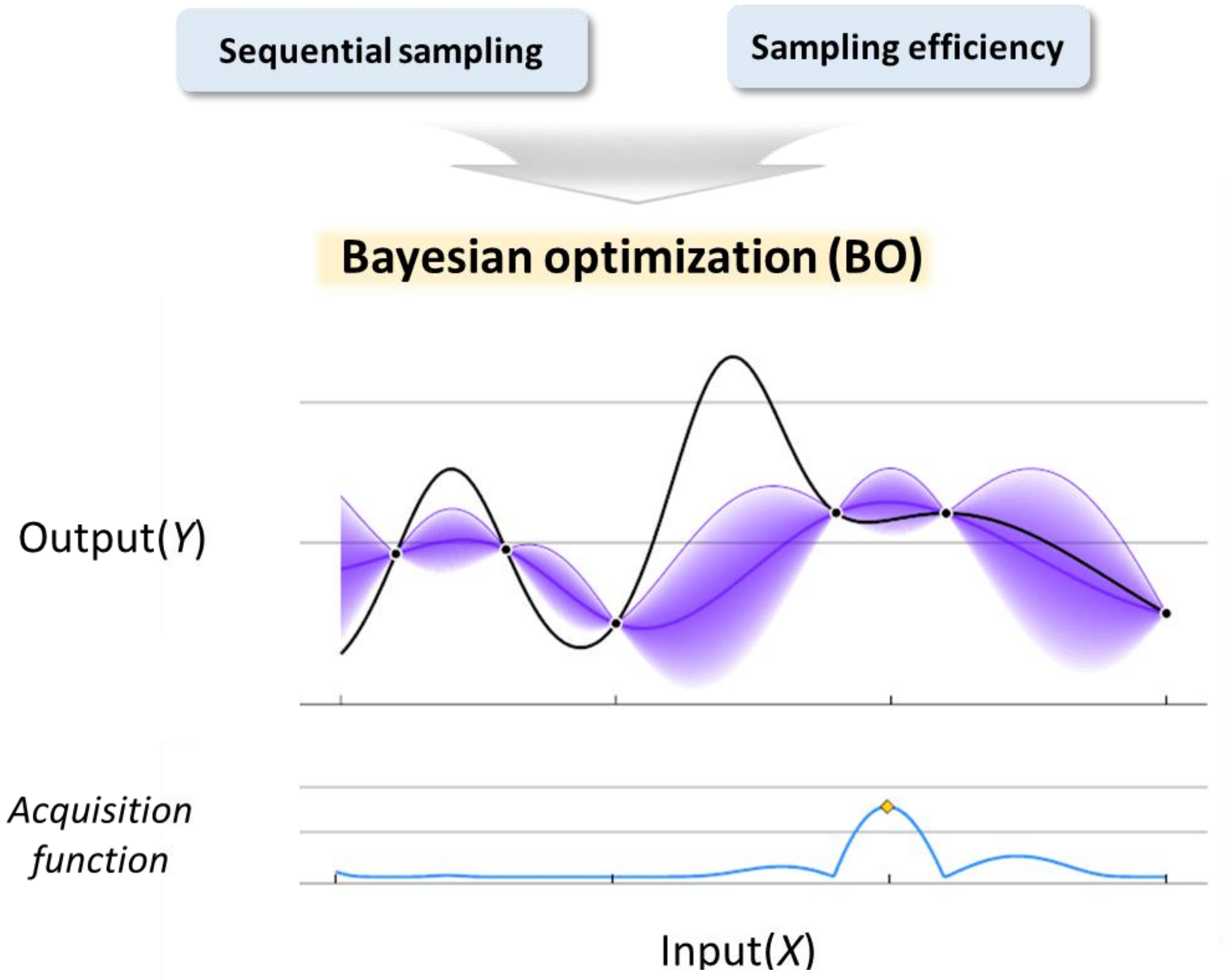

$$\mathbf{x}_\star = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$$

- Many decision-making problems in engineering domains can be cast as black-box optimization problems

- Where $\boldsymbol{f}(\boldsymbol{x})$ **is a black-box**, i.e.
  - ✓ We may only be able to observe the function value (**no gradients**)
  - ✓ Typically, **sampling is expensive**

Source: Jones et. al., "Efficient global optimization of expensive black-box functions." Journal of Global Optimization. (1998)

# Bayesian optimization



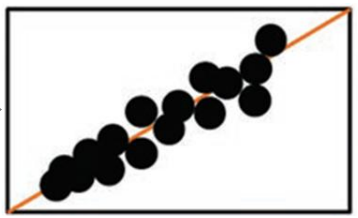

- ❖ An **interactive decision-making** strategy for global optimization of black box functions

- ❖ Balancing between **exploration and exploitation** by utilizing uncertainty estimates
  - ➢ Surrogate model is constructed from the data
  - ➢ Based on the model, acquisition function suggests the next experiment input

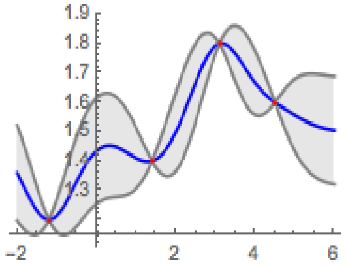
# Surrogate model - Gaussian process (GP)

- ❖ Most common surrogate model for Bayesian optimization
- ❖ GP provides **not only mean, but also confidence** of estimates

*Surrogate model, f(X)*

Input **X** →  → **Y** Output

**Gaussian Process Model (GPM)**



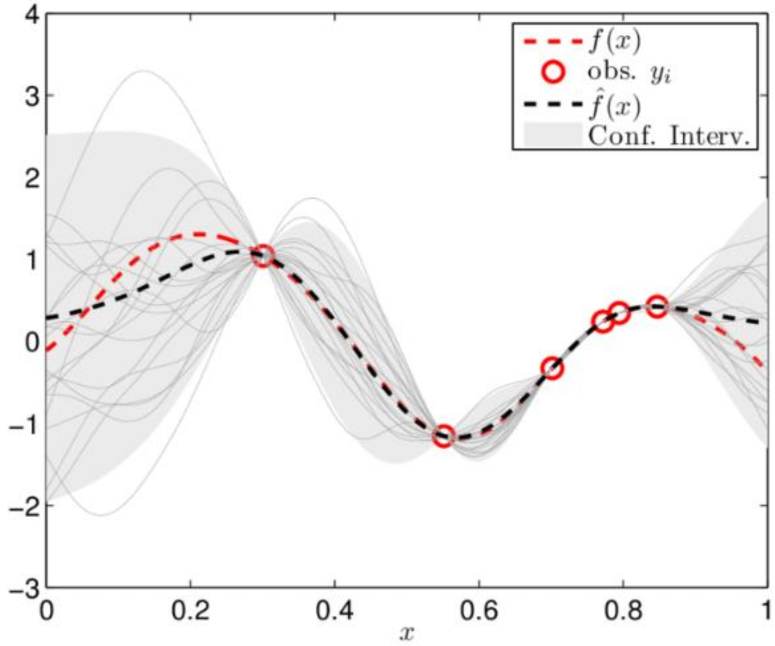




*Figure 1. Example of Gaussian process model*

Source: Wolfram, Hadid et al. Design of low-cost sensors for industrial processes energy consumption measurement. *Sensing Technology*, 23–46 (2015).

# Acquisition function

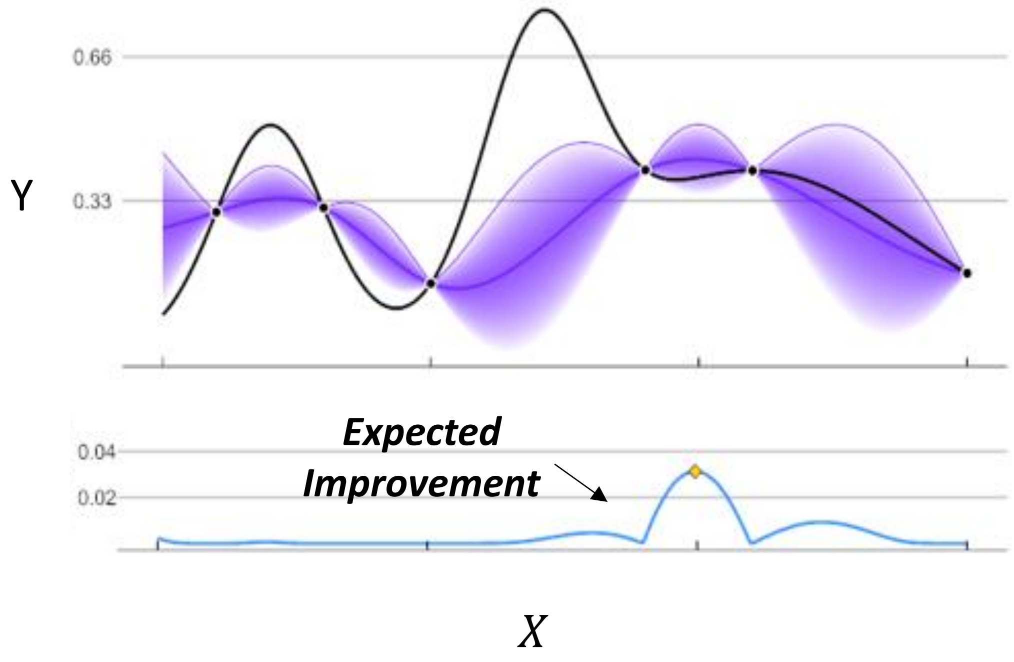



*Figure 2. Illustration of acquisition function (EI)*

- ❖ Acquisition function guides where to sample at next trial, $t+1$
- ❖ Choice of acquisition function determines a way to balance between **exploration and exploitation**
- ❖ As an example, **Expected Improvement (EI)** is the most popular acquisition function

$$u(x) = max\left(f(x) - f^{\text{best}}, 0\right)$$

$$a_{\text{EI}}(x) = \mathbb{E}[u(x) \mid x, \mathcal{D}_{\text{t}}]$$

# Problem with conventional Bayesian optimization

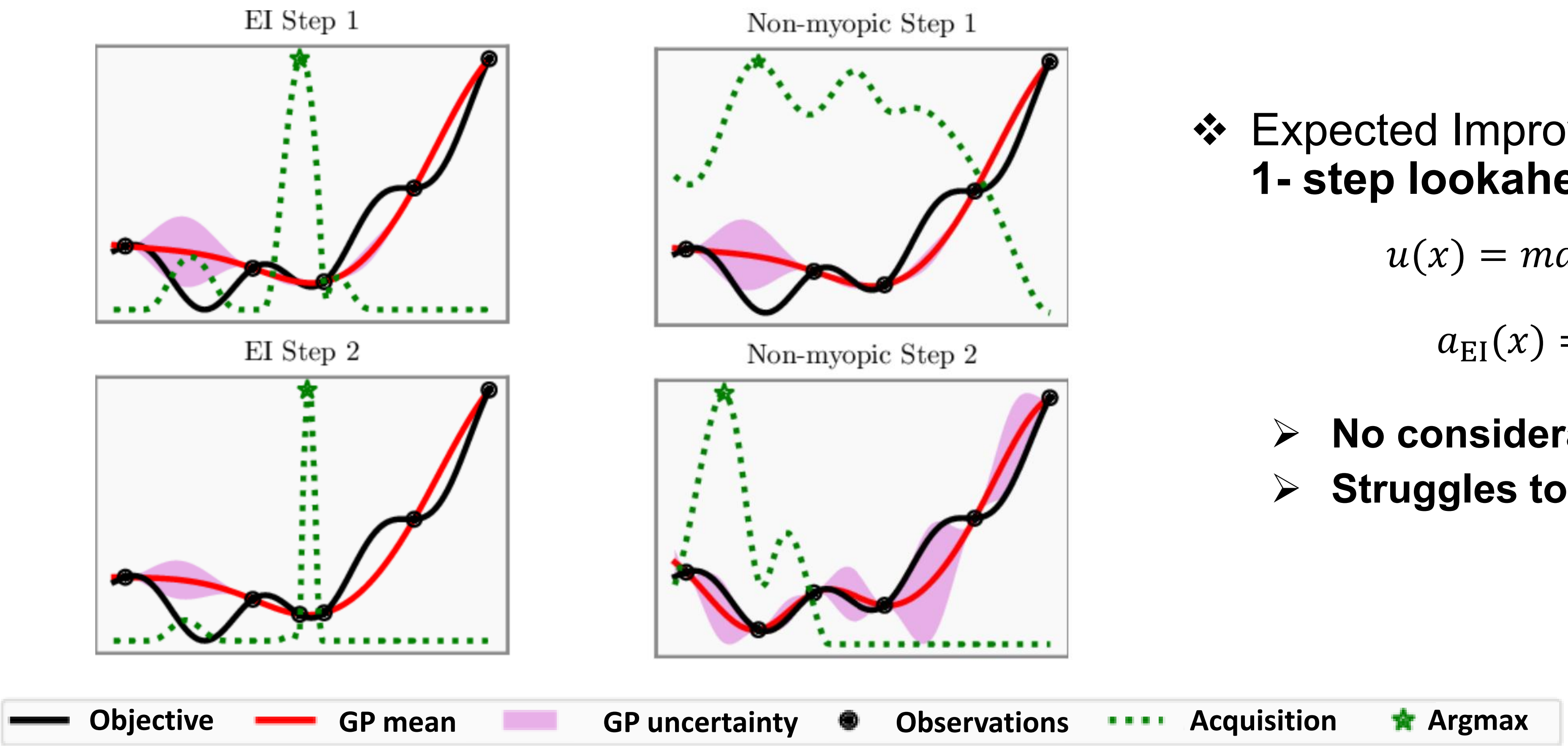



*Figure 3. Illustration of acquisition function values*

- Expected Improvement only cares about **1- step lookahead** decision making

$$u(x) = max\left(f^{\text{best}} - f(x), 0\right)$$

$$a_{\text{EI}}(x) = \mathbb{E}[u(x) \mid x, \mathcal{D}_t]$$

  - **No consideration of future decisions**
  - **Struggles to escape local minima**

Source: Lee et. al., "Efficient rollout strategies for bayesian optimization." Conference on Uncertainty in Artificial Intelligence. PMLR. (2020)

# Lookahead Bayesian optimization as a dynamic program

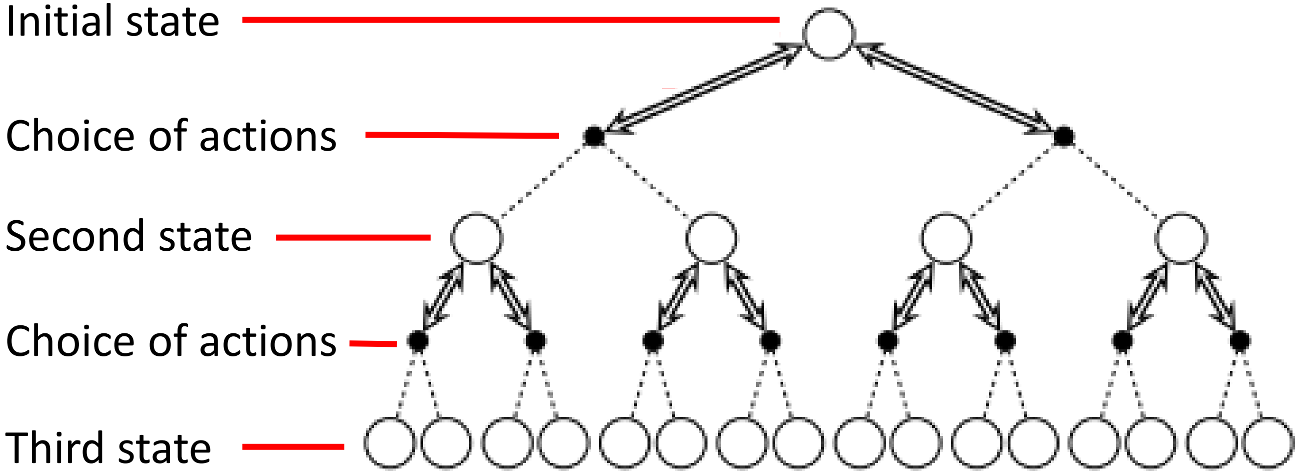



*Figure 4. Illustration of Dynamic programming*

- ❖ Lookahead BO can be expressed as a dynamic program (DP)
  - ➢ Decision at time $t$ influence decisions in time $t+1$
- ❖ Solving DP is computationally **extremely heavy**
  - ➢ Rollout based BO has been introduced by Lam et al.

Source: Lam, Remi, Karen Willcox, and David H. Wolpert. "Bayesian optimization with a finite budget: An approximate dynamic programming approach." NIPS. (2016)

# Rollout based BO

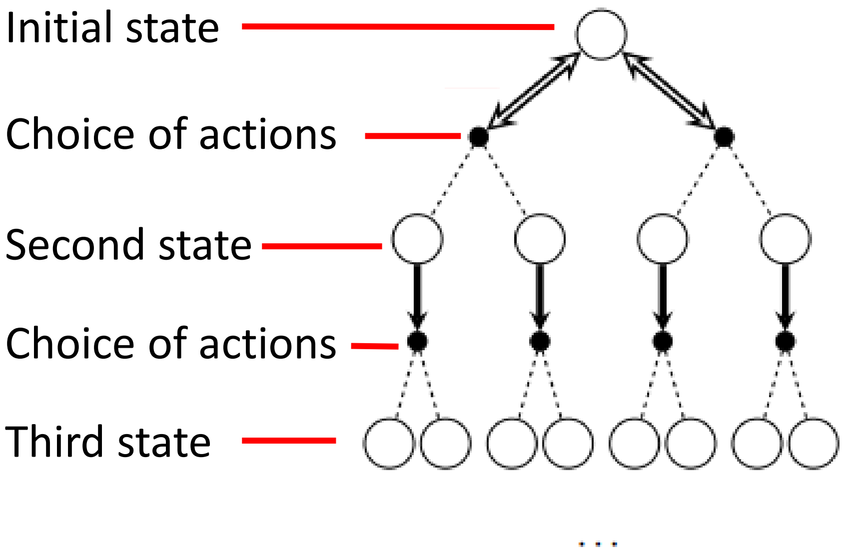



*Figure 5. Illustration of Rollout based BO*

- To mitigate this computational expense, Rollout based BO has been suggested
  - For the 1st step, actions are optimized as normal
  - For the 2nd ~ $h^{th}$ decision, a **heuristic policy** (such as EI) is applied
  - **No freedom** of choice from the 2nd decision

Source: Lam, Remi, Karen Willcox, and David H. Wolpert. "Bayesian optimization with a finite budget: An approximate dynamic programming approach." NIPS. (2016)

# Proposed approach: Reinforcement learning based BO

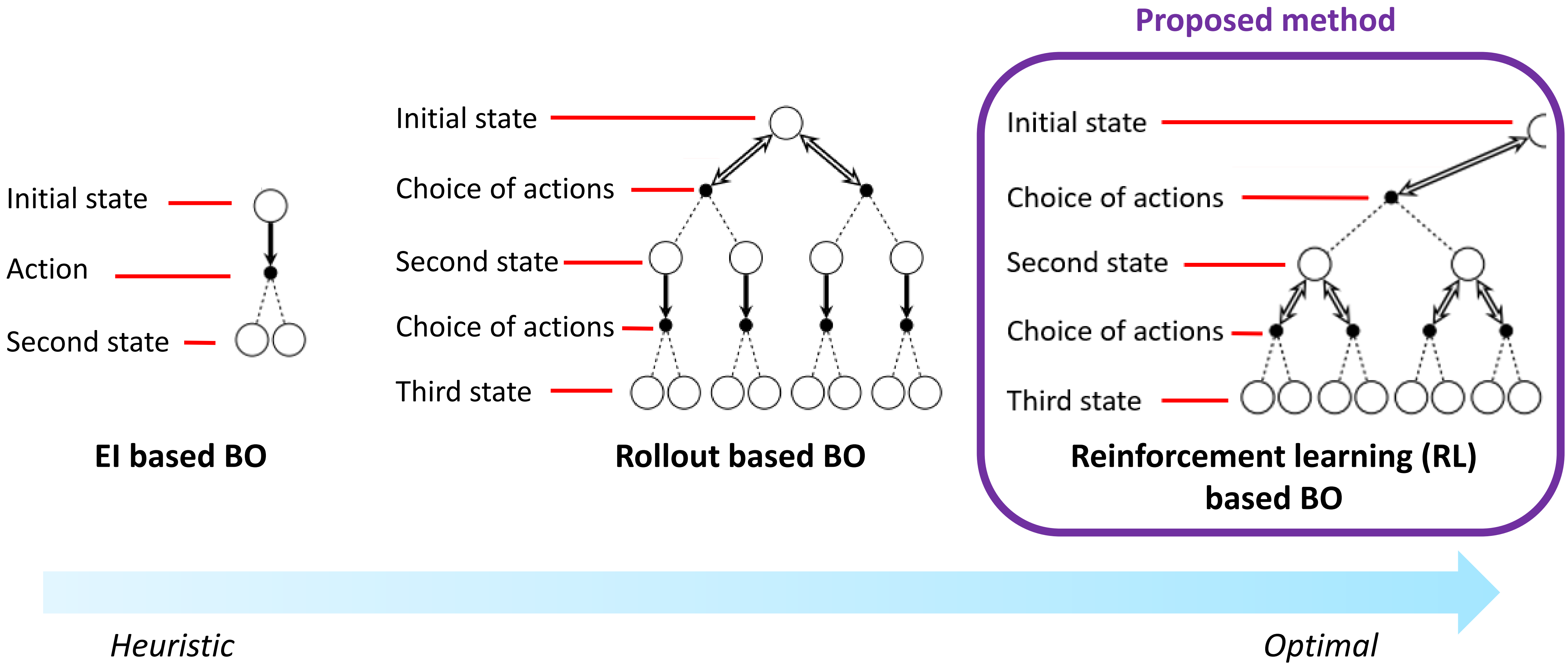



*Figure 6. Illustration of spectrum of BO*

# Reinforcement learning based BO

- **Reinforcement learning (RL)**: a method to learn about the optimal decision on a certain state
- On a certain state, RL agent makes an action and receives reward from the environment
- RL can **solve DP in a near optimal way**

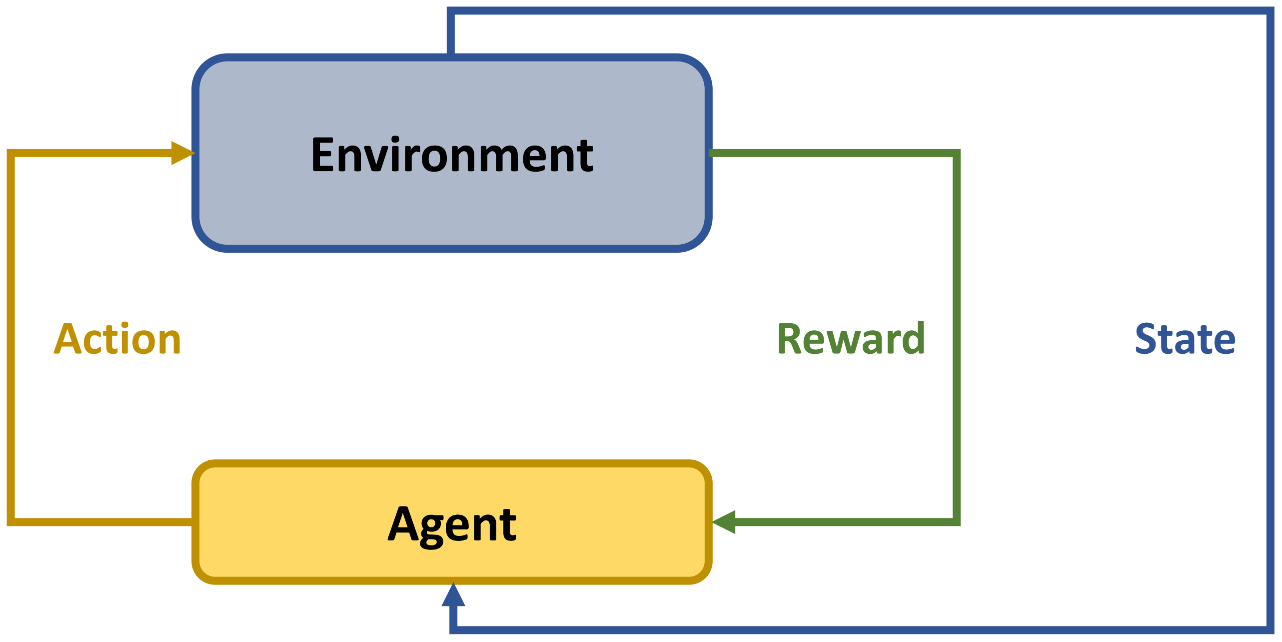



*Figure 7. Principle of Reinforcement learning*

Source: Barto, Andrew G. "Reinforcement Learning: An Introduction. By Richard's Sutton." SIAM Rev 6.2 (2021): 423.

# Dyna architecture

- Learn a model from real experience
- **Learn and plan** value function from real and simulated experiences

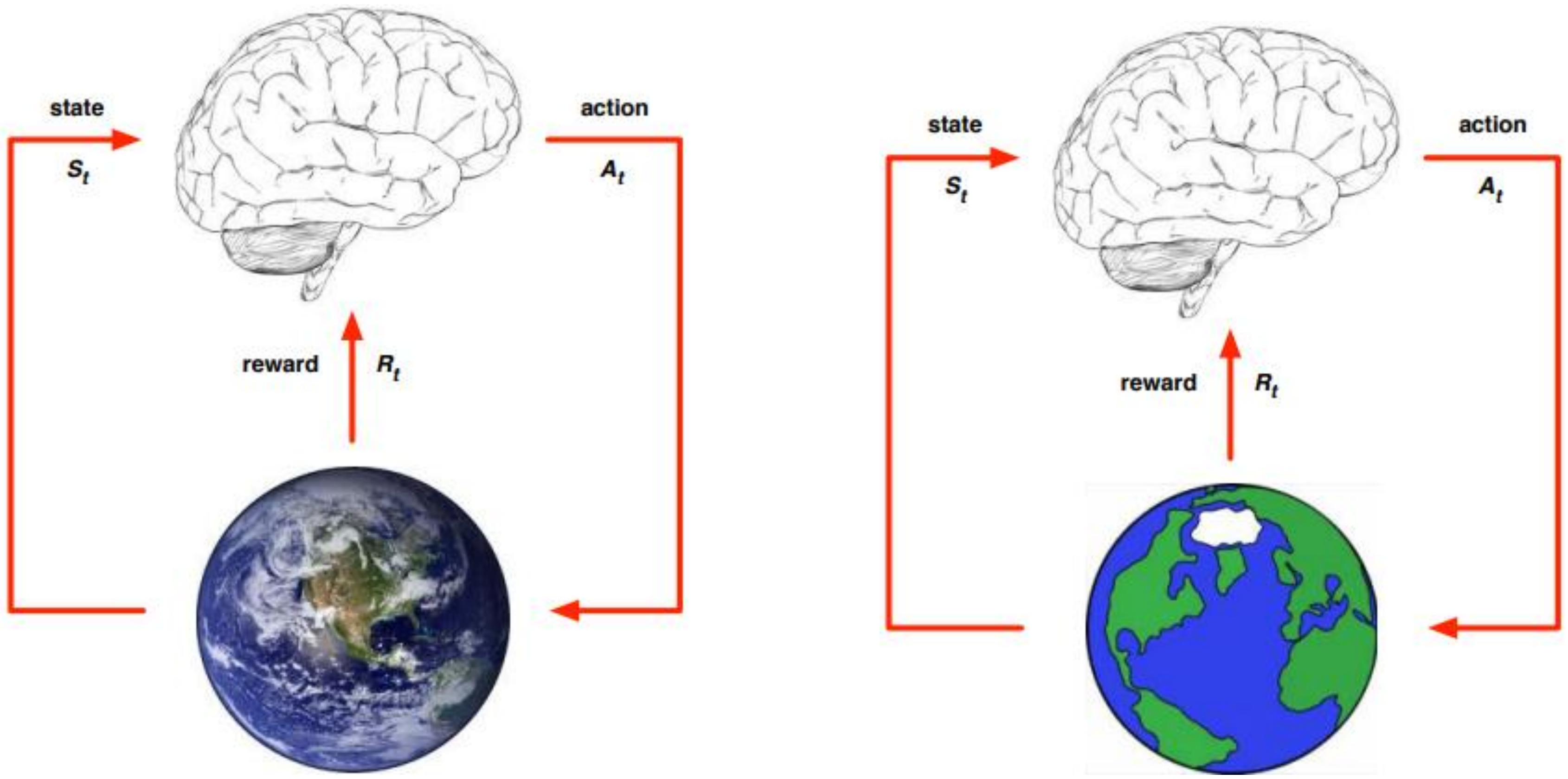



*Figure 8. Illustration of two different RL methods*

Source: David Silver lecture

# Proposed state space – Encoder-based representation

❖Attention - and DeepSets-based Neural Network for RL-BO

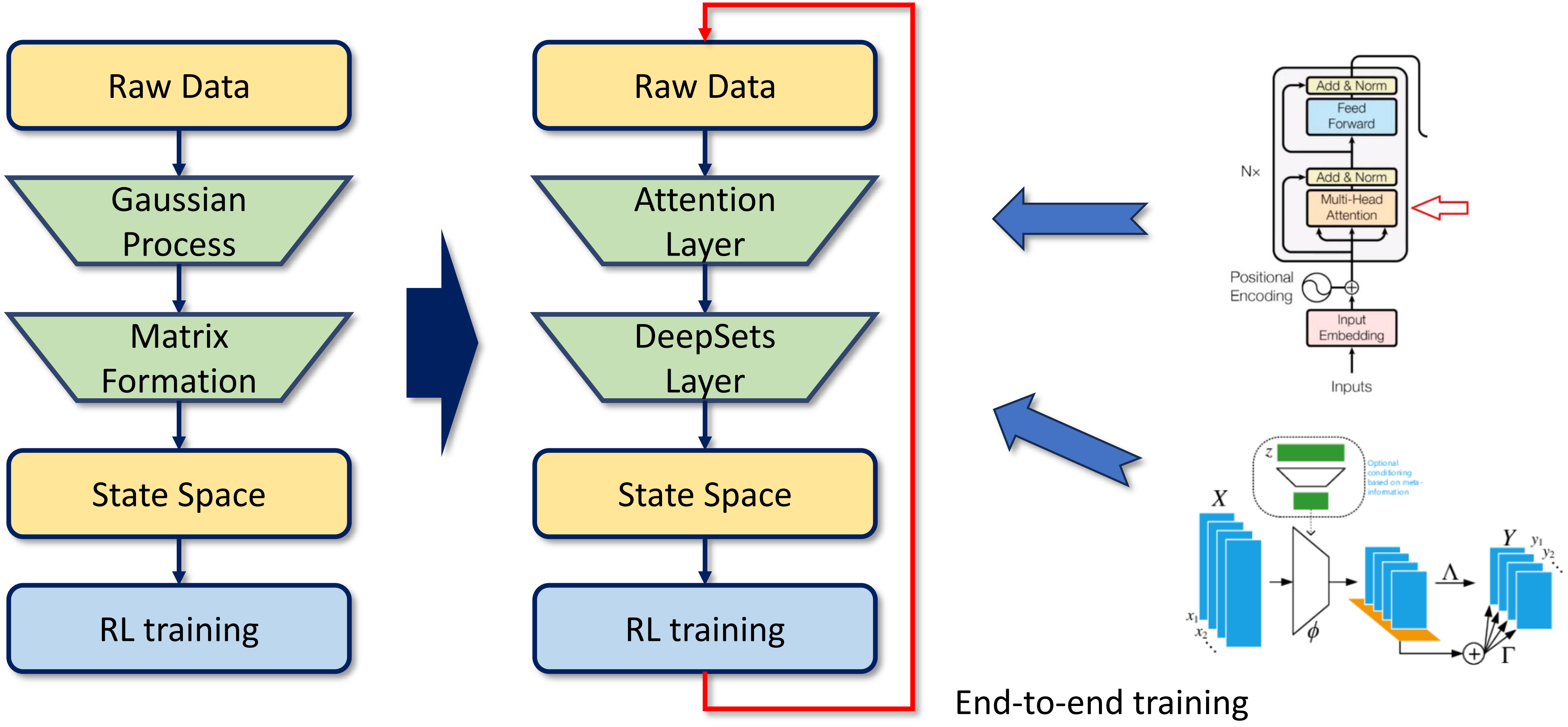



*Figure 9. Illustration for the encoder-based state representation*

# EARL-BO (Encoder Augmented RL for Bayesian Optimization)

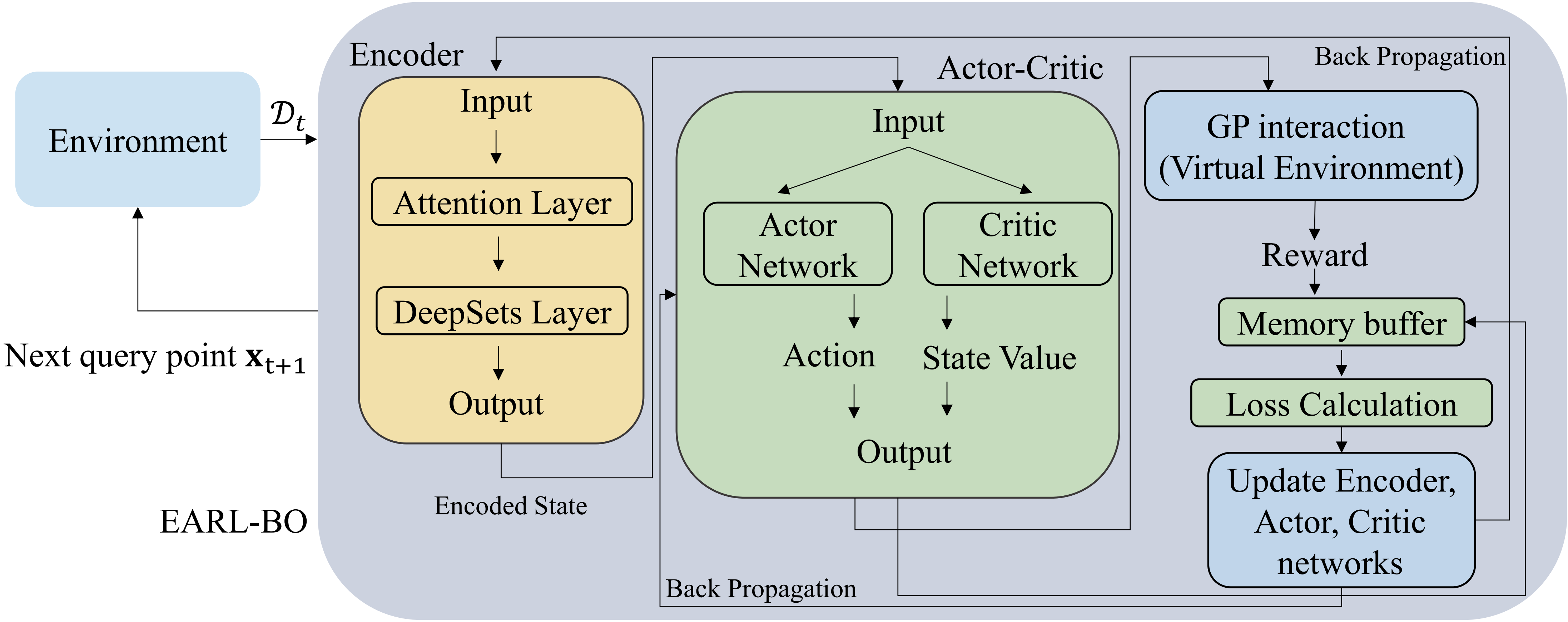


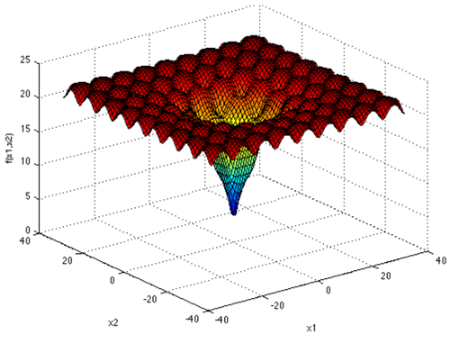

*Figure 10. An overview of the EARL-BO architecture*

Source: Cheon, Mujin, et al. "EARL-BO: Reinforcement Learning for Multi-Step Lookahead, High-Dimensional Bayesian Optimization." arXiv preprint arXiv:2411.00171 (2024).

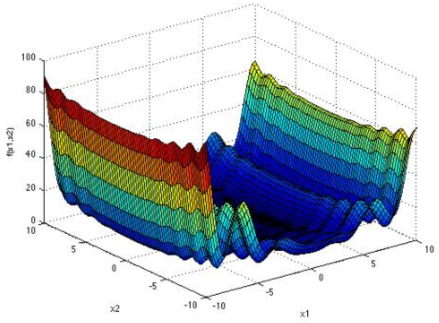
# Case study

❖ Four case studies with different benchmark functions with different dimension (2D, 5D, 8D, 30D)

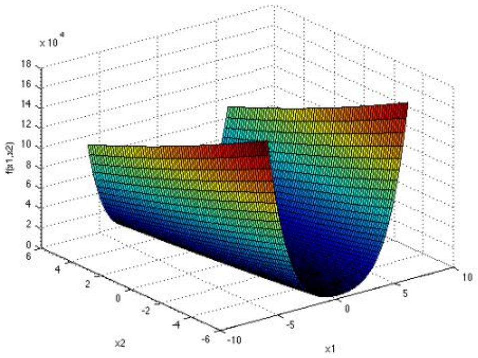
*Ackley function*



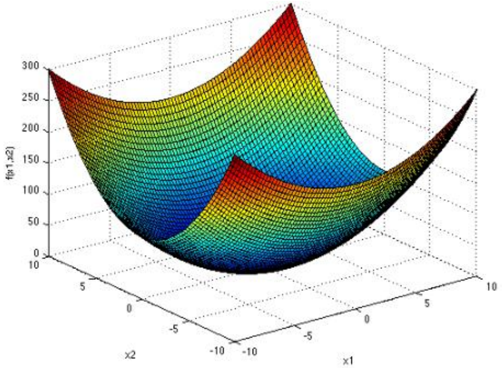
*Levy function*



*Rosenbrock function*



*Sum squares function*



❖ Three case studies with HPO-B benchmark data from OpenML (6D, 8D, 19D)

❖ Compared with Random, EI, Rollout-BO, TuRBO, and SAASBO as a benchmark

❖ Performance index

- **Regret** = the difference between the optimal value and the best point in dataset at time $t$ was recorded for the performance comparison (i.e. $y_{opt} - y_t^*$)
- Averaged over 10 experiments

Source: Cheon, Mujin, et al. "EARL-BO: Reinforcement Learning for Multi-Step Lookahead, High-Dimensional Bayesian Optimization." arXiv preprint arXiv:2411.00171 (2024).

# Case study: Benchmark functions

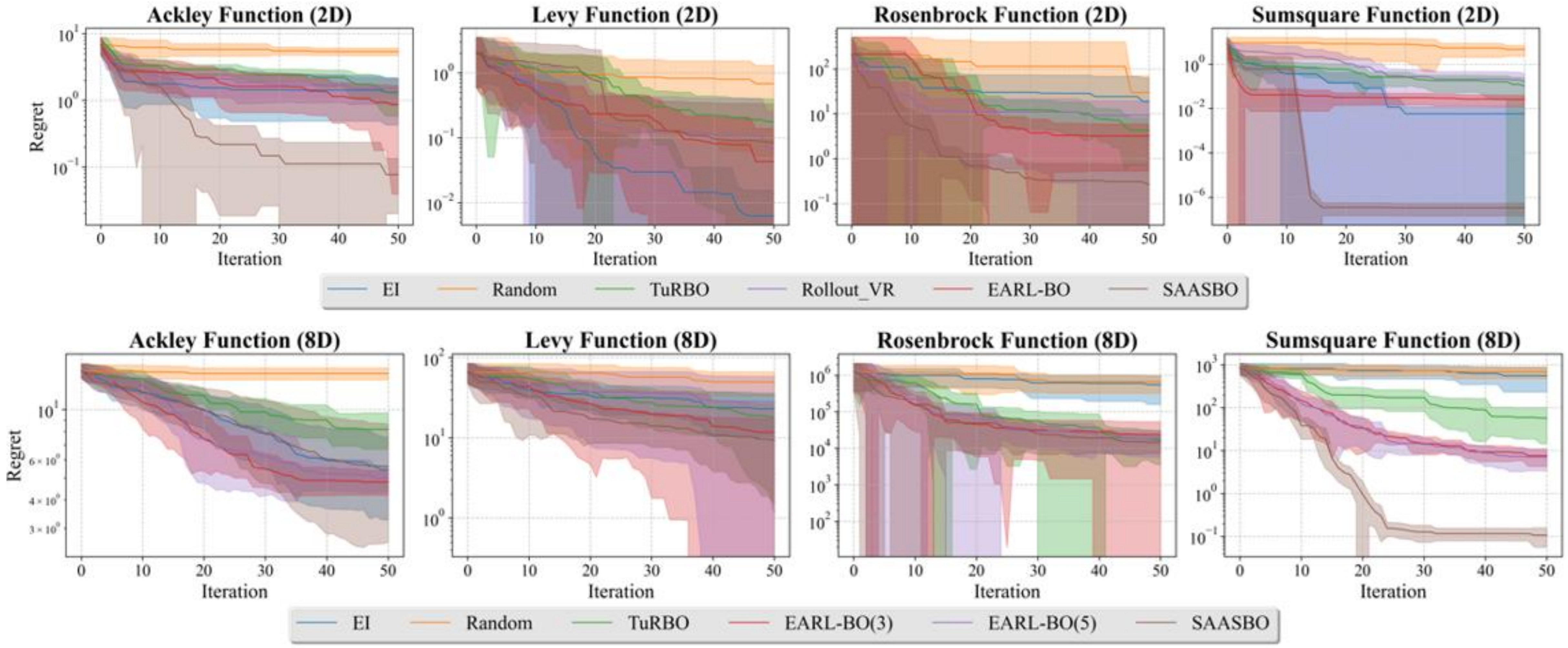



*Figure 11. Optimization performance on 2D, 8D benchmark functions*

# Case study: Benchmark functions

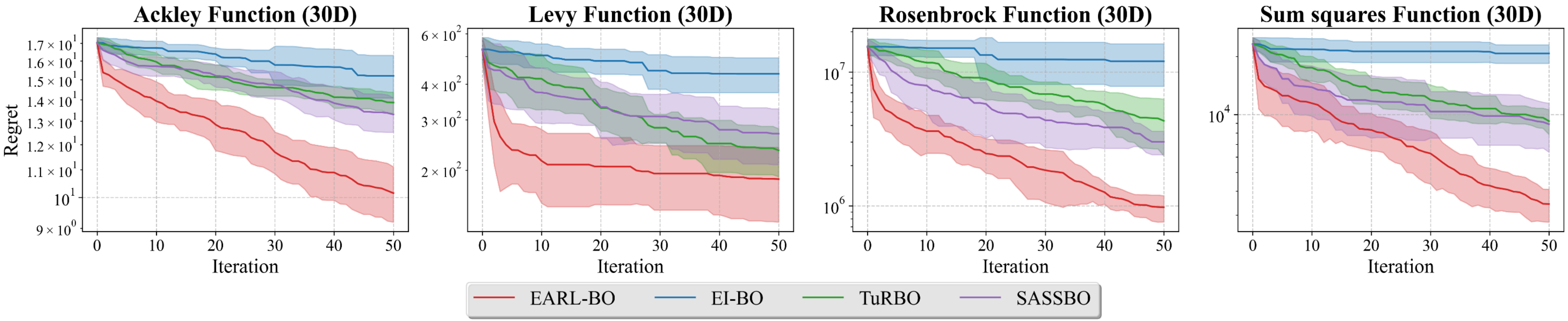



*Figure 12. Optimization performance on 30D benchmark functions*

Source: Cheon, Mujin, et al. "EARL-BO: Reinforcement Learning for Multi-Step Lookahead, High-Dimensional Bayesian Optimization." arXiv preprint arXiv:2411.00171 (2024).

# Case study: Hyperparameter optimization (HPO-B data)

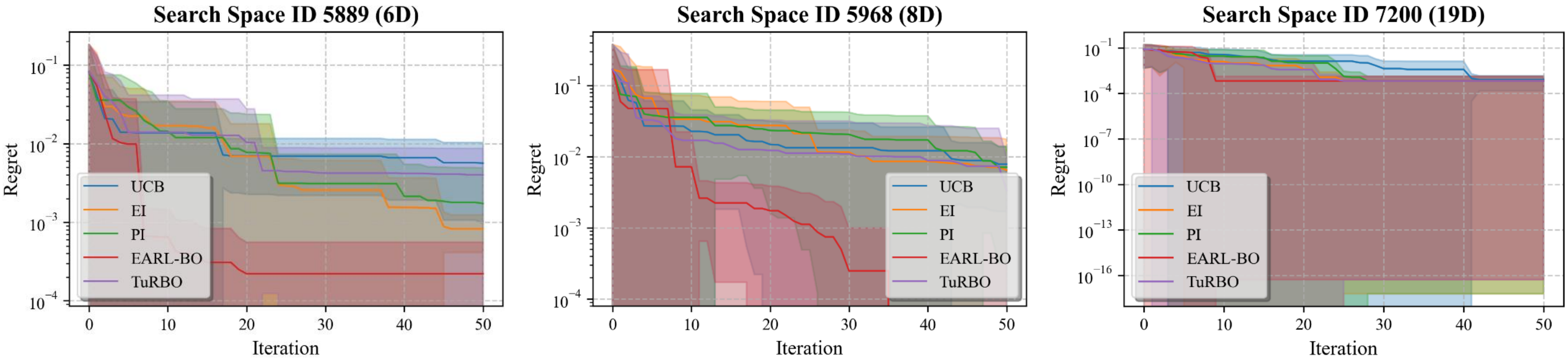



*Figure 13. Optimization performance on various Hyperparameter optimization problems*

Source: Cheon, Mujin, et al. "EARL-BO: Reinforcement Learning for Multi-Step Lookahead, High-Dimensional Bayesian Optimization." arXiv preprint arXiv:2411.00171 (2024).

# Conclusion

❖ **EARL-BO shows superior optimization performance compared to existing rollout-based BO and high-dimensional BO methods in various dimensions**

- Implementation of encoder-based RL could be a way of making non-myopic and RL-based BO to be applicable for high-dimensional BO
- However, it takes long time (~850s in PC) to make 1-step decision due to computational load

ICML paper number: 9113

arXiv link to the paper

Session 4, Topic: Zero-order and Black-box Optimization

# Thank you for your attention

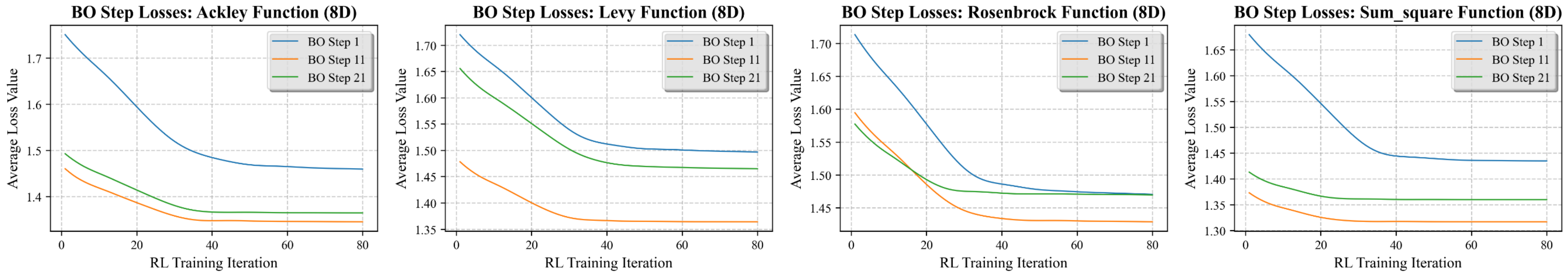
ICML 2025

Presenter LinkedIn

Authors: Mujin Cheon, Dong Yeun Koh, Jay H. Lee, and Calvin Tsay

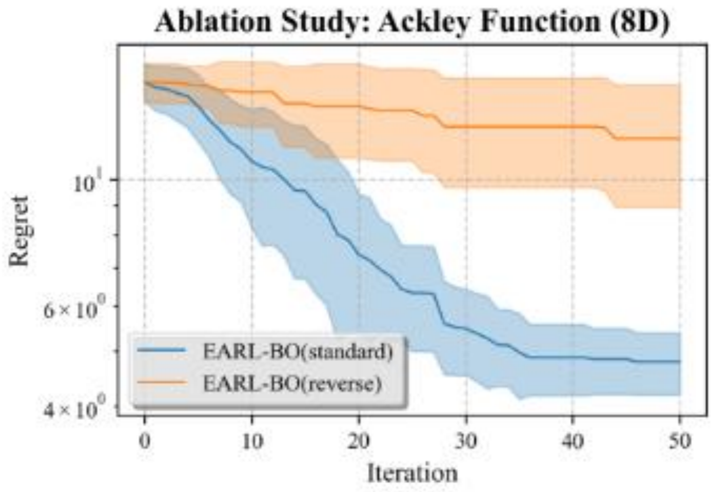
# Additional results

## ❖ Learning rate of RL




## ❖ Difference in learning rate between RL and Encoder

- ➢ Standard RL and encoder modules are, respectively, (0.001, 0.01)



# Ablation study

What happens if we use sequence as a state? (without the permutation invariant)

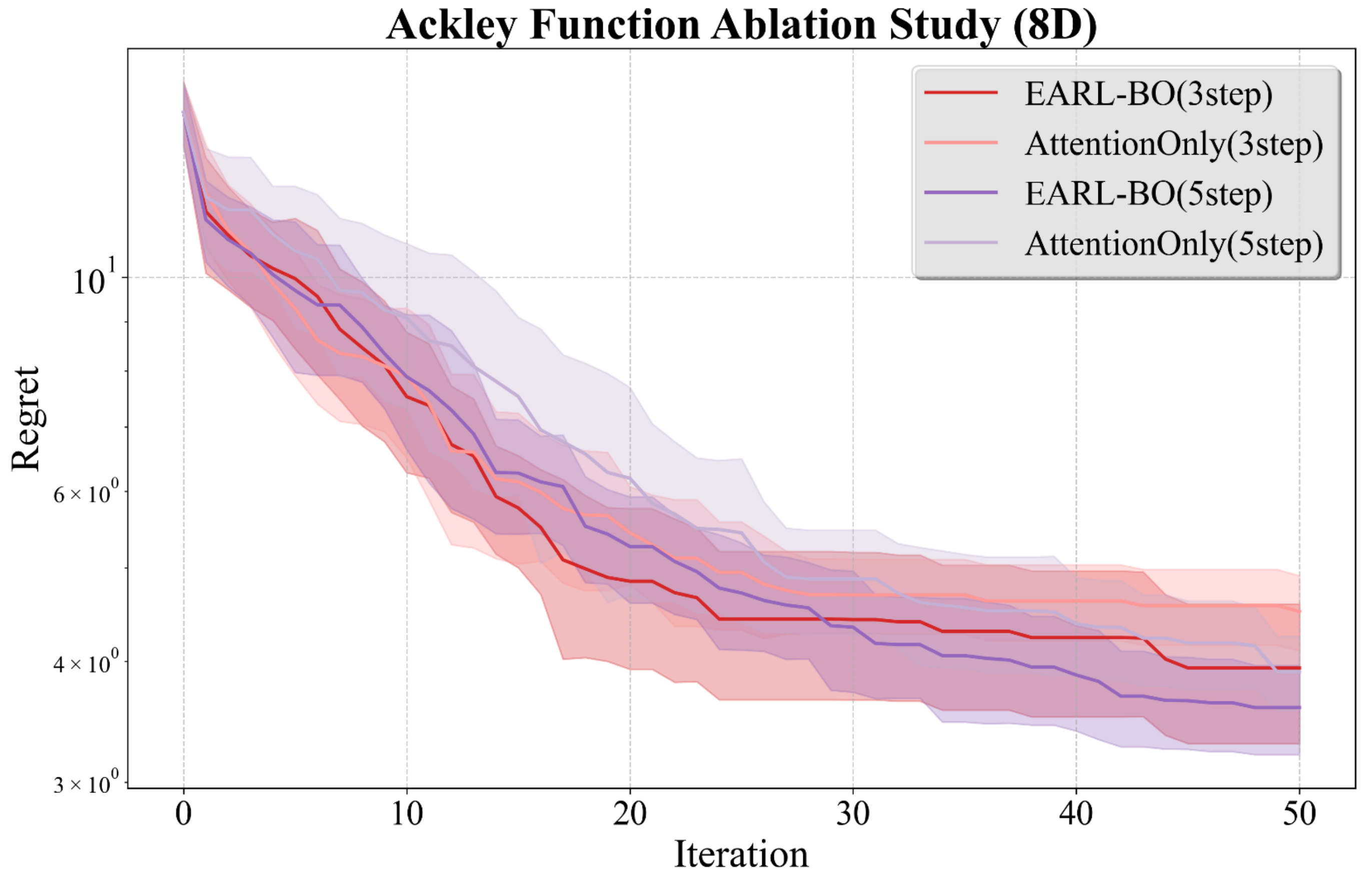
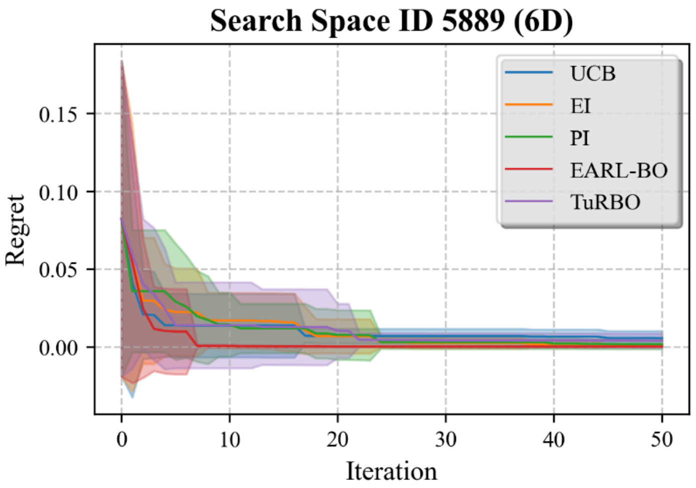
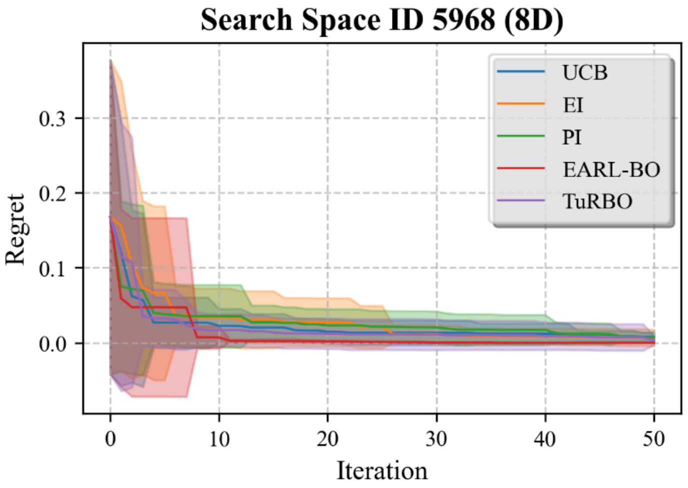


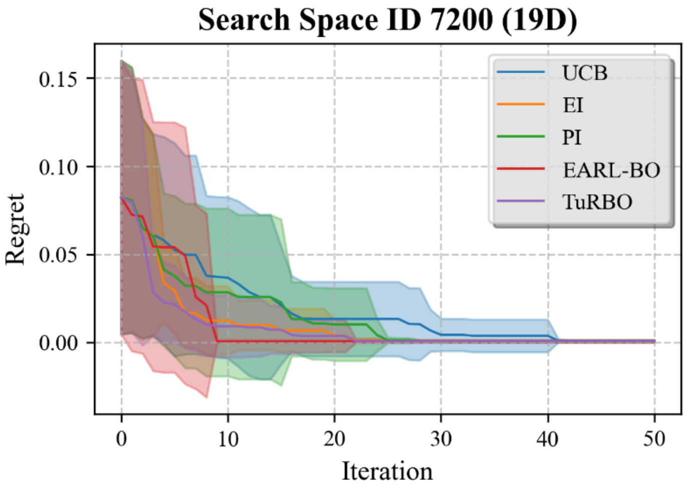

*Figure 16. Optimization performance on various benchmark functions*

# Additional results

## ❖ Scale of standard deviation




## ❖ Planning delusion