# Residual Kernel Policy Network: Enhancing Stability and Robustness in RKHS-Based Reinforcement Learning

*Yixian Zhang, Huaze Tang, Huijing Lin, Wenbo Ding*

Tsinghua Shenzhen International Graduate School, Tsinghua University

**The Thirteenth International Conference on Learning Representations (ICLR 2025)**

Reporter: Yixian Zhang

Supervisor: Professor Wenbo Ding

Tsinghua University

# Context

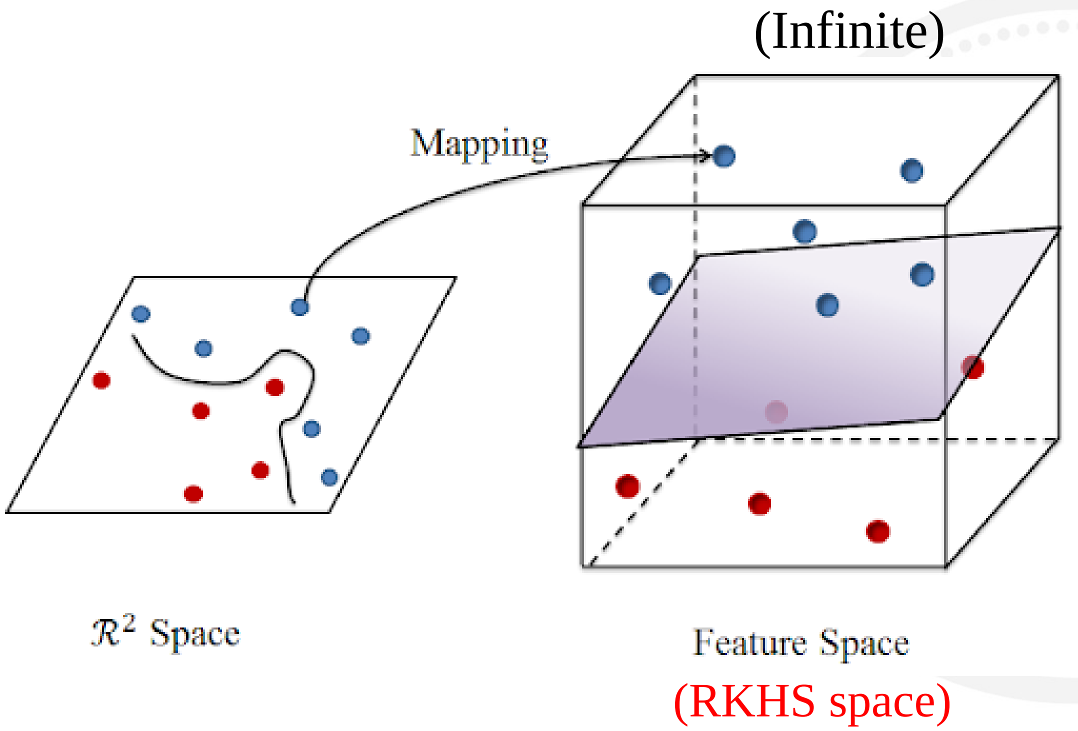
# Reproducing Kernel Hilbert Space (RKHS)



Reproducing Kernel Hilbert Space (RKHS) is the vector valued Hilbert Space $H_K$ where an elements $K(x,\cdot) \in H$ satisfies the reproducing property $\langle K(x,\cdot), K(y,\cdot)\rangle = K(x,y)$.
(RKHS is dependent on the kernel $K(x,\cdot)$ of it !)

- Kernel mapping: $K(x,\cdot)$, mapping vector $x$ in $\mathbb{R}$ into the RKHS. (which is usually infinite)

- Universal kernel: The kernel is universal when $\{f \mid f = \sum c_i K(x_i,\cdot)\} = \mathbb{C}(\mathbb{R})$, that means the functions in $H_K$ can approximate any continuous functions.



# RKHS Policy Gradient



**Policy**: Adopting the Gaussian policy, choose the action from a multivariate normal distribution $N(h_\omega(s), \Sigma)$: ($h$ is parameterized by $\omega$)

$$\pi_{h_\omega, \mathbf{\Sigma}}(a|s) := \frac{1}{Z} e^{-\frac{1}{2}(h_\omega(s)-a)^\top \mathbf{\Sigma}^{-1}(h_\omega(s)-a)}$$

$h \rightarrow h_\omega$

**RKHS policy**: Model $h$ directly in RKHS: ($h \in H_k$)

$$\pi_{h, \mathbf{\Sigma}}(a|s) := \frac{1}{Z} e^{-\frac{1}{2}(h(s)-a)^\top \mathbf{\Sigma}^{-1}(h(s)-a)}$$

The RKHS policy can be updated directly through RKHS gradient:

$$\nabla_h \hat{U}(\pi_h)(\cdot) = \eta K(s_k, \cdot) \mathbf{\Sigma}^{-1}(a_k - h(s_k)) \hat{Q}^{\pi_h}(a_k, s_k)$$

$$h = h + \nabla_h \hat{U}(\pi_h)$$

# Why RKHS Policy not Work in Complex Env.

**(a) Robustness/ Representation**: Different environments need different kernel hyperparameters. Meanwhile, the policy can also benefit from **representation learning in state**.

**(b) Variance**: A main factor influencing the performance of policy gradient algorithms. In RKHS policy, we find that the use of **kernel increase the variance greatly**.

# Why RKHS Policy not Work in Complex Env.

**(a) Robustness/ Representation**: Different environments need different kernel hyperparameters. Meanwhile, the policy can also benefit from **representation learning in state**.

**(b) Variance**: A main factor influencing the performance of policy gradient algorithms. In RKHS policy, we find that the use of **kernel increase the variance greatly**.

**Our contributions:**

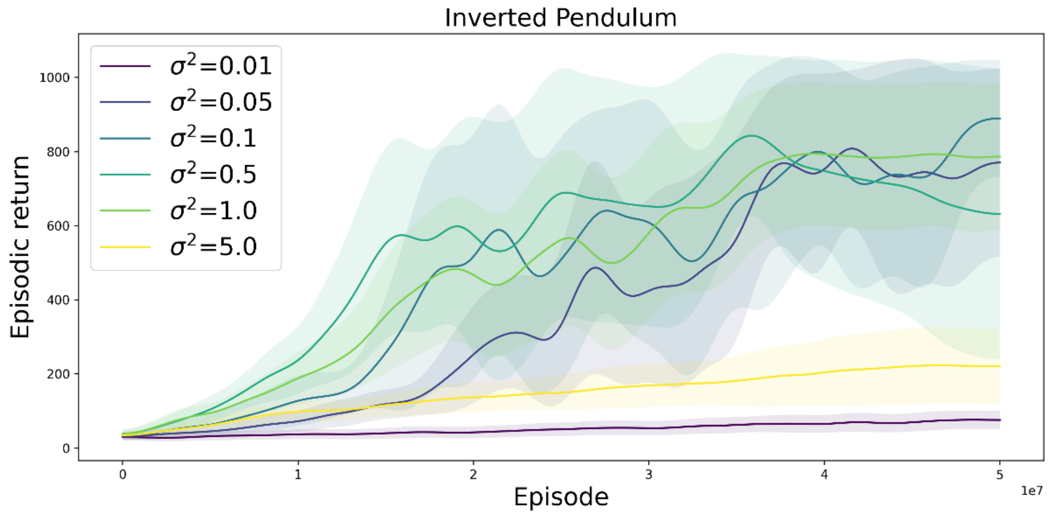
- We analyze the high variance issue in RKHS policy gradient, which leads to significant instability and high variance during training
- (To solve (a)): We propose Kernel Policy Network (KPN), integrating RKHS and neural network by aligning observation distributions with the chosen kernel.
- (To solve (b)): We propose ResKPN, combining the residual layer to effectively reduce the variance in RKHS policy gradient.

# Analysis for RKHS Policy

TBSI 清华-伯克利深圳学院 Tsinghua-Berkeley Shenzhen Institute

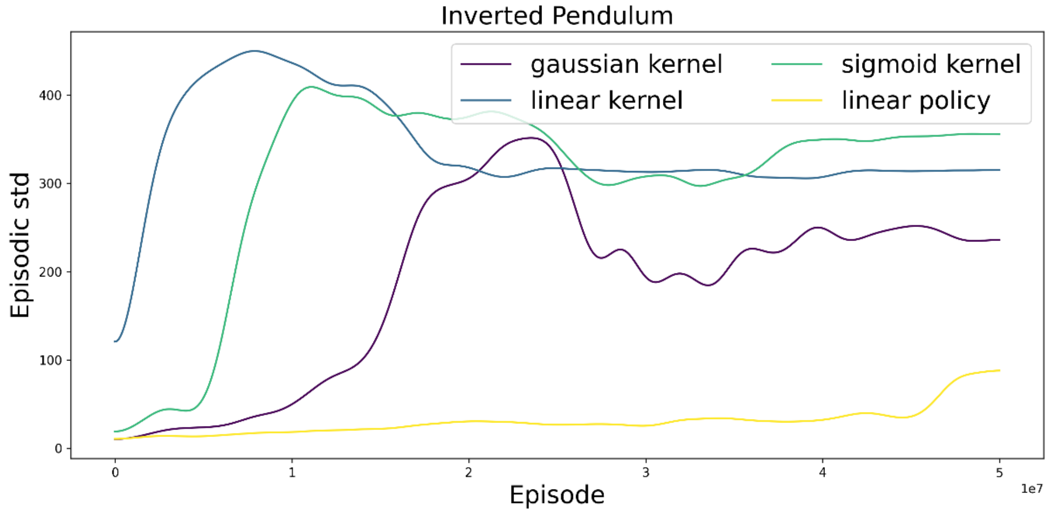
**Insufficient representational capacity:**

Different hyperparameters influence the learning performance significantly




**Excessive variance:**

The use of kernel influences the variance in training significantly (all kernel policy got a high variance)



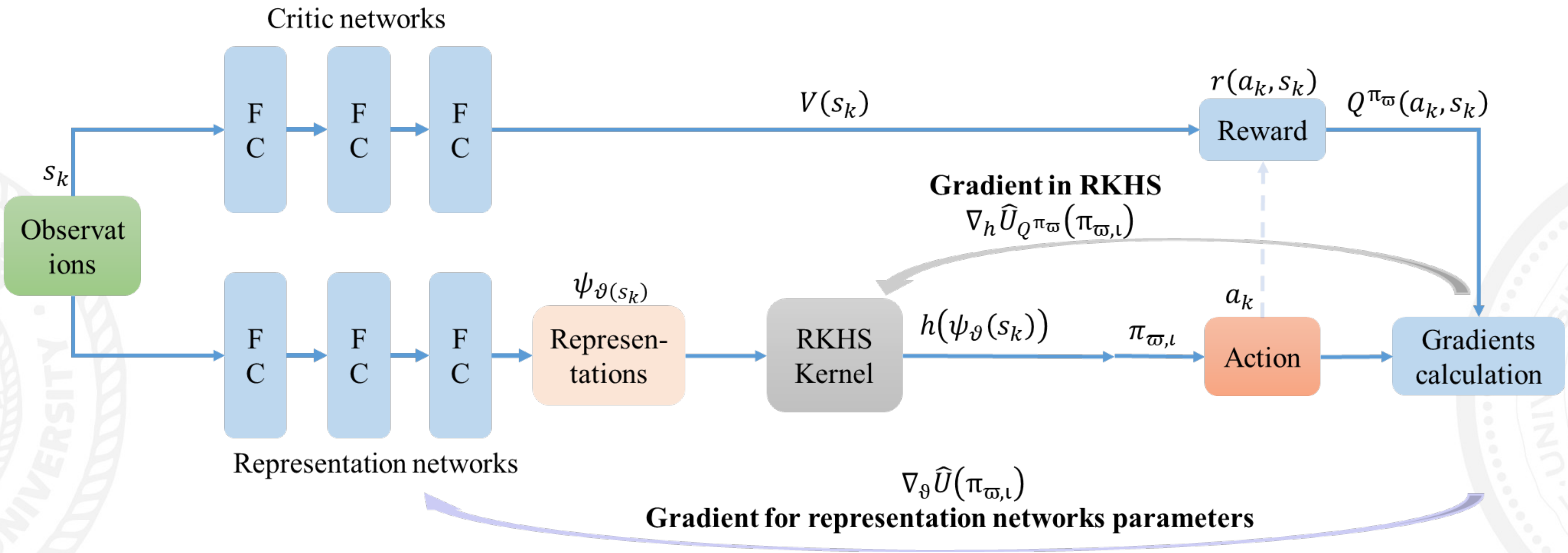

**Lemma 1:**

$$\mathbb{E}_{s_k}\left[\frac{\mathrm{Var}_{a_k}\left(\nabla_h \hat{U}\left(\pi_h\right)\right)}{\mathrm{Var}_{a_k}\left(\nabla_\theta \hat{U}\left(\pi_\theta\right)\right)}\right] \geq \mathbb{E}_{s_k}\left[\frac{K^2(s_k, s_k)}{s_k^2}\frac{2c_1^2}{2c_2^2+\Sigma^{-1}(c_2\theta^\top s_k+d_2)^2}\right].$$

# Representation Learning for RKHS Policy

TBSI 清华-伯克利深圳学院 Tsinghua-Berkeley Shenzhen Institute

## KPN (Kernel Policy Network)



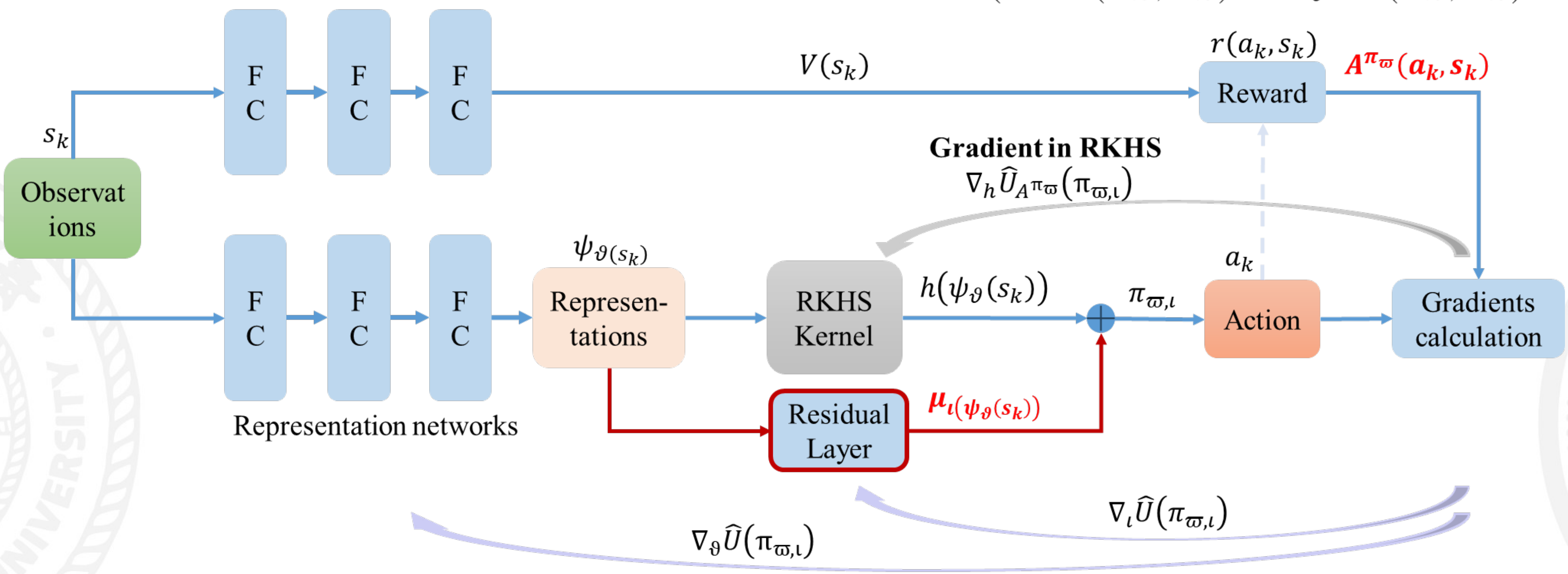

$$\nabla_h \hat{U}(\pi_h) = \eta K(s_k, \cdot)\boldsymbol{\Sigma}^{-1}(a_k - h(s_k))\hat{Q}^{\pi_h}(a_k, s_k) \longrightarrow \eta K(\psi_\vartheta(s_k), \cdot)\boldsymbol{\Sigma}^{-1}(a_k - h(\psi_\vartheta(s_k)))Q^{\pi_\varpi}(a_k, s_k)$$

Use neural network $\psi_\vartheta(s_k)$ to learn representations for state $s_k$, and the Critic network is utilized.

# Variance Reduction for RKHS Policy

## ResKPN (Residual Kernel Policy Network)

$$\left(A^{\pi_\varpi}(a_k, s_k) = Q^{\pi_\varpi}(a_k, s_k) - V^{\pi_\varpi}(s_k)\right)$$




**Gradient for representation networks parameters**

$$\eta K(\psi_\vartheta(s_k), \cdot)\Sigma^{-1}(a_k - h(\psi_\vartheta(s_k)))Q^{\pi_\varpi}(a_k, s_k) \longrightarrow \eta K(\psi_\vartheta(s_k), \cdot)\Sigma^{-1}(a_k - h(\psi_\vartheta(s_k)) - \mu_\iota(\psi_\vartheta(s_k)))A^{\pi_\varpi}(a_k, s_k)$$

The advantage function $A^{\pi_\varpi}(a_k, s_k)$ is used, and residual layer $\mu_\iota(\psi_\vartheta(s_k))$ is added to stabilize the training (also help representation learning)

# Analysis for Variance Reduction

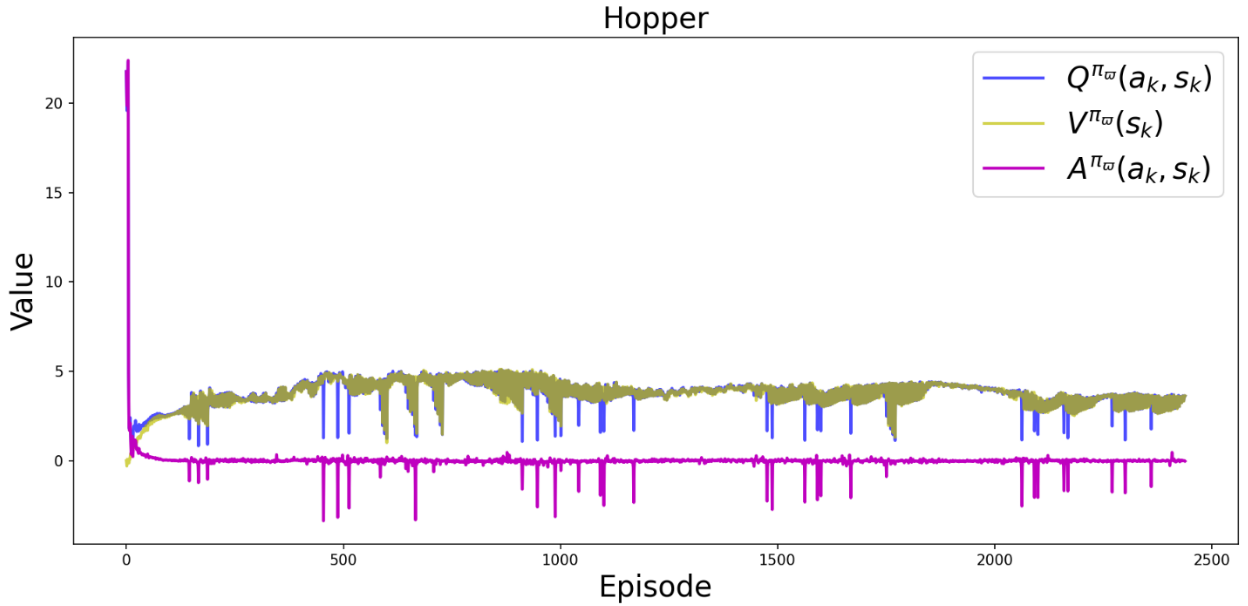


**Theorem 1:** $\mathrm{Var}_{a_k}(\nabla_h \hat{U}(\pi_\varpi)) \geq \mathrm{Var}_{a_k}(\nabla_h \hat{U}_{A^{\pi_\varpi}}(\pi_\varpi)) \geq \mathrm{Var}_{a_k}(\nabla_h \hat{U}_{A^{\pi_\varpi}}(\pi_{\varpi,\iota}))$

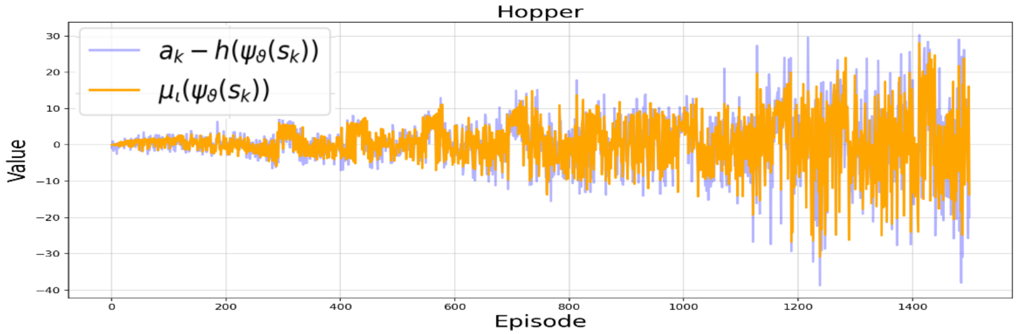
**Why** the Theorem 1 work? We display these value respectively for an intuitive explanation:

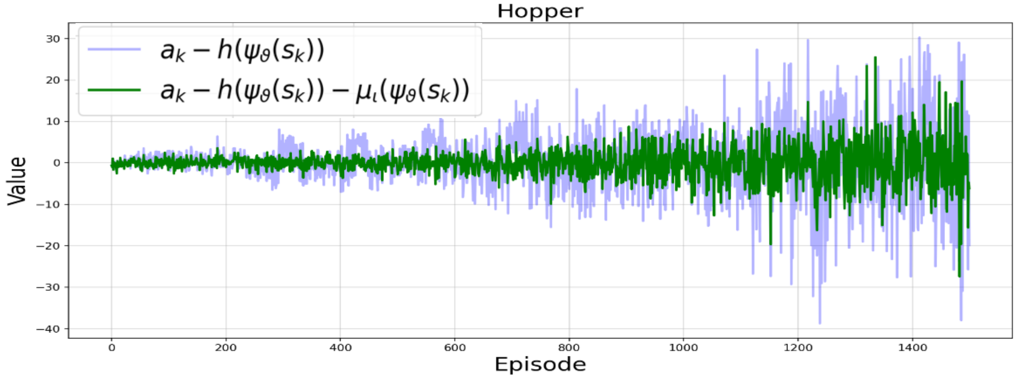
Recalling the gradient: $\eta K(\psi_\vartheta(s_k), \cdot)\boldsymbol{\Sigma}^{-1}(a_k - h(\psi_\vartheta(s_k)) - \mu_\iota(\psi_\vartheta(s_k)))A^{\pi_\varpi}(a_k, s_k)$

For advantage function $A^{\pi_\varpi}(a_k, s_k)$ $(A = Q - V)$




For residual layer $\mu_\iota(\psi_\vartheta(s_k))$



# Analysis for Variance Reduction

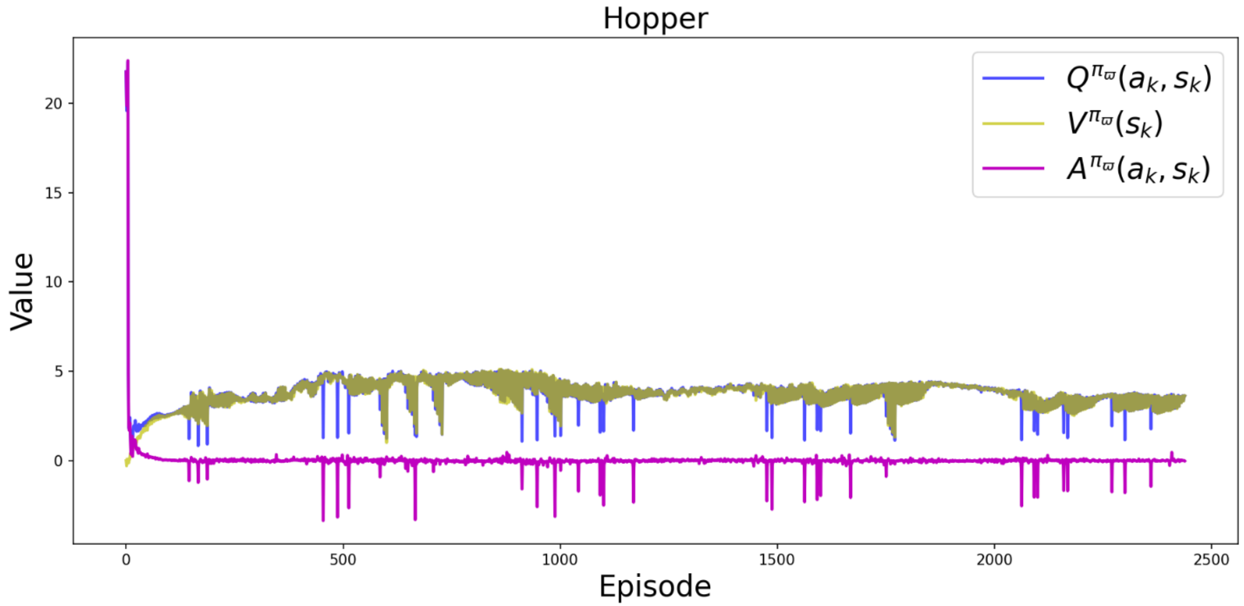
TBSI 清华-伯克利深圳学院 Tsinghua-Berkeley Shenzhen Institute

**Theorem 1:** $\mathrm{Var}_{a_k}(\nabla_h \hat{U}(\pi_\varpi)) \geq \mathrm{Var}_{a_k}(\nabla_h \hat{U}_{A^{\pi_\varpi}}(\pi_\varpi)) \geq \mathrm{Var}_{a_k}(\nabla_h \hat{U}_{A^{\pi_\varpi}}(\pi_{\varpi,\iota}))$

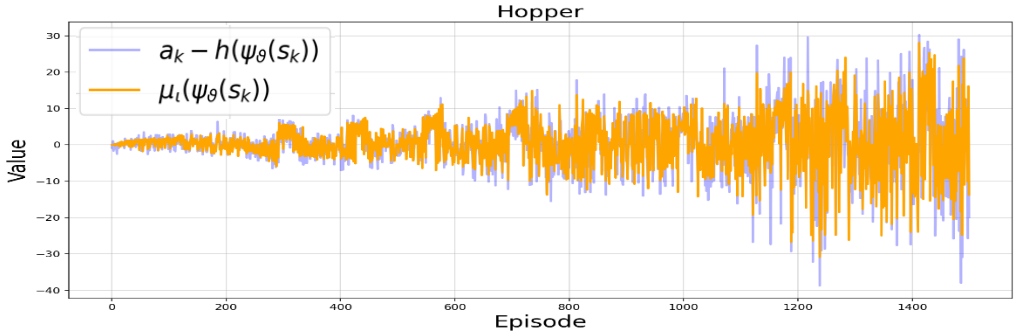
**Why** the Theorem 1 work? We display these value respectively for an intuitive explanation:

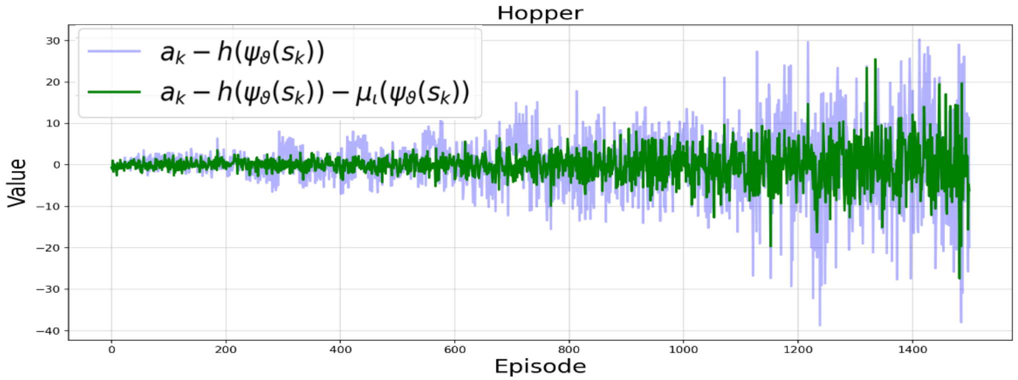
Recalling the gradient: $\eta K(\psi_\vartheta(s_k),\cdot)\boldsymbol{\Sigma}^{-1}(\boxed{a_k - h(\psi_\vartheta(s_k))} - \boxed{\mu_\iota(\psi_\vartheta(s_k))})A^{\pi_\varpi}(a_k, s_k)$

For advantage function $A^{\pi_\varpi}(a_k, s_k)$ $(A = Q - V)$




For residual layer $\mu_\iota(\psi_\vartheta(s_k))$



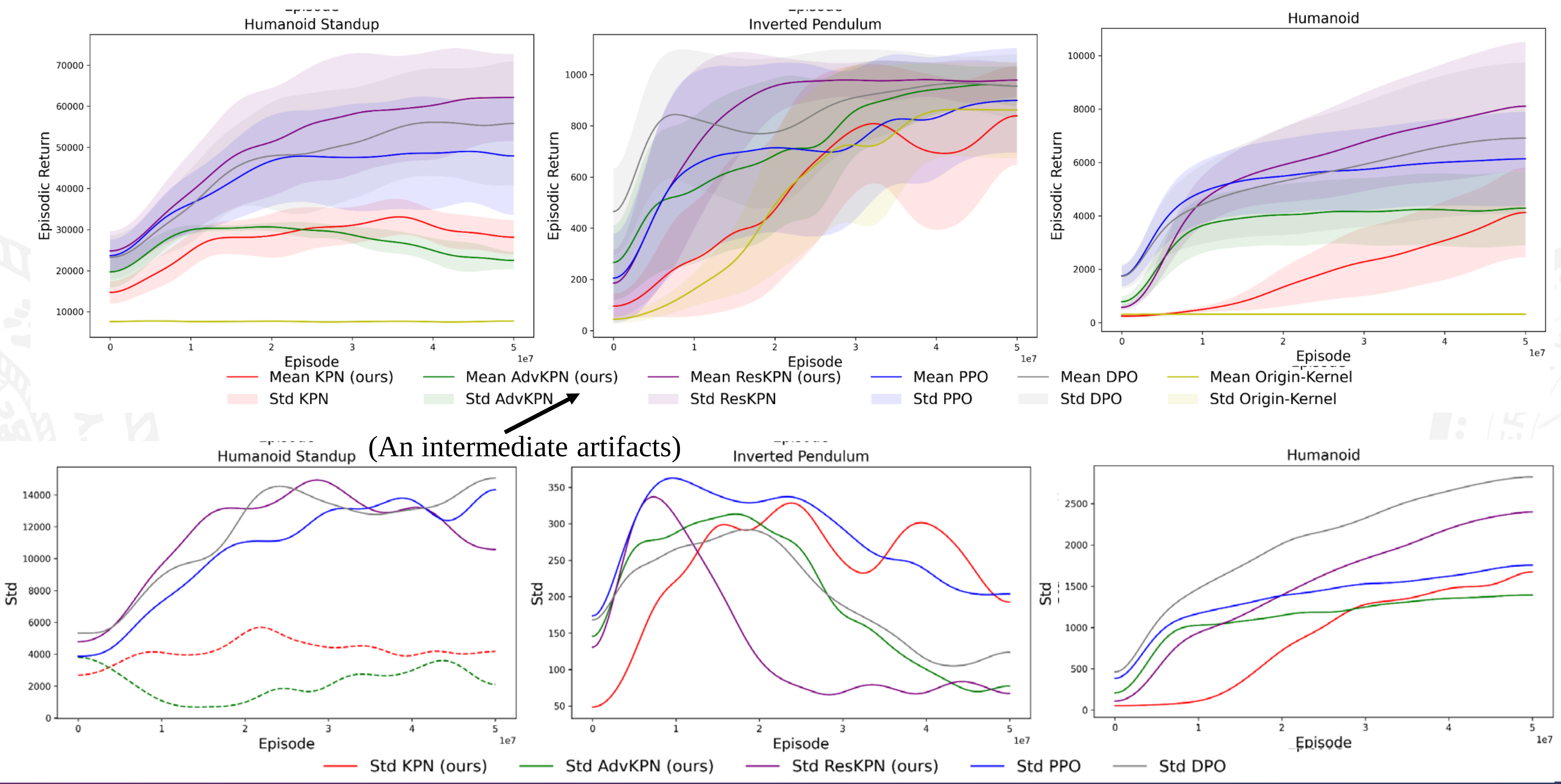
# Experiment Context



- Overall performance in episodic reward and training variance
- Analysis for the effectiveness

# Overall Performance





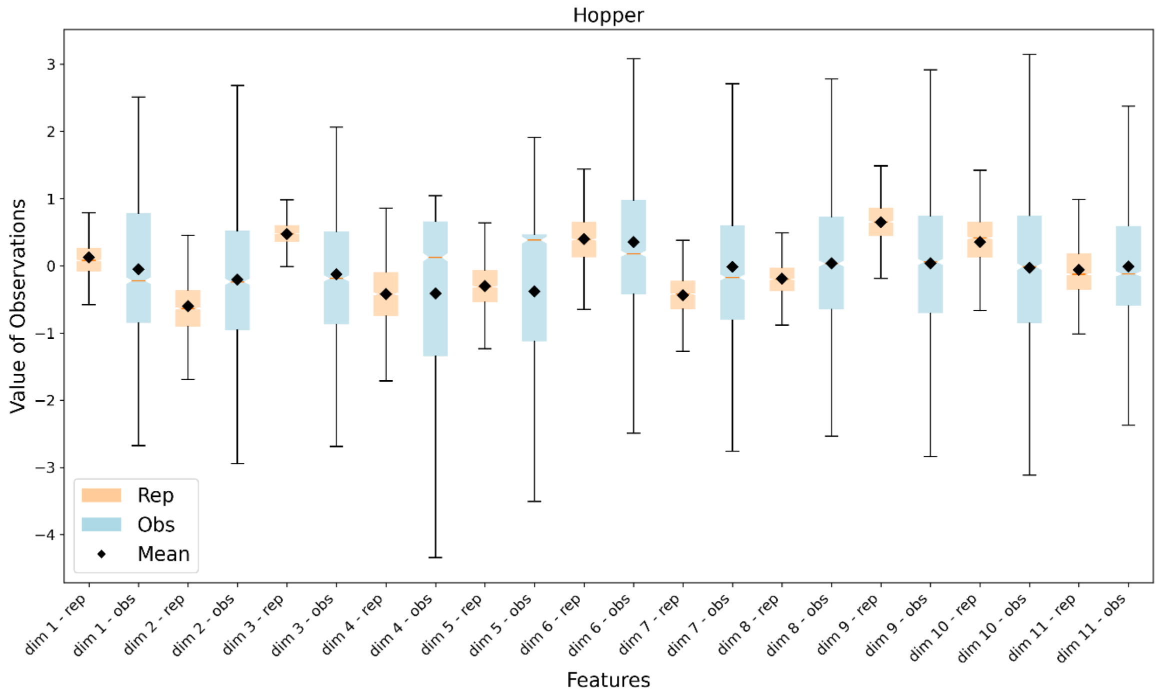
# Effectiveness Analysis

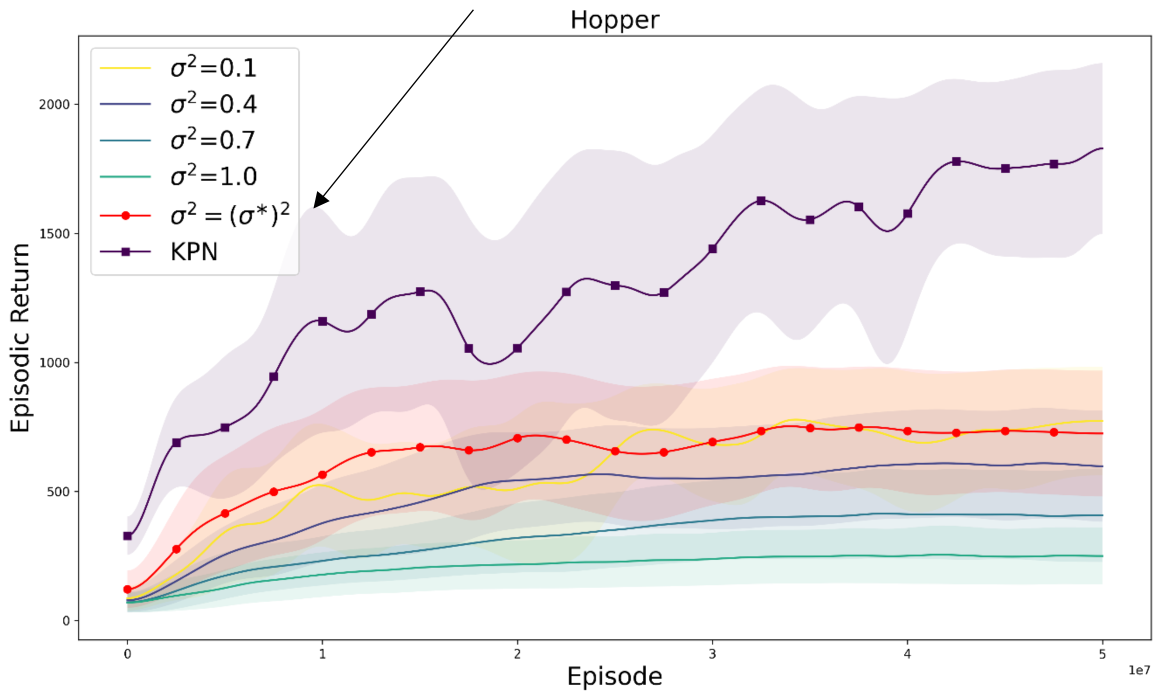
TBSI 清华-伯克利深圳学院 Tsinghua-Berkeley Shenzhen Institute

**For the representation learning:** The representation learning is adjusting the distribution of the state.

If we adjust the hyper parameter $\sigma$ to align the distribution in representation learning, will the performance increase?

Best parameter to adjust the distribution



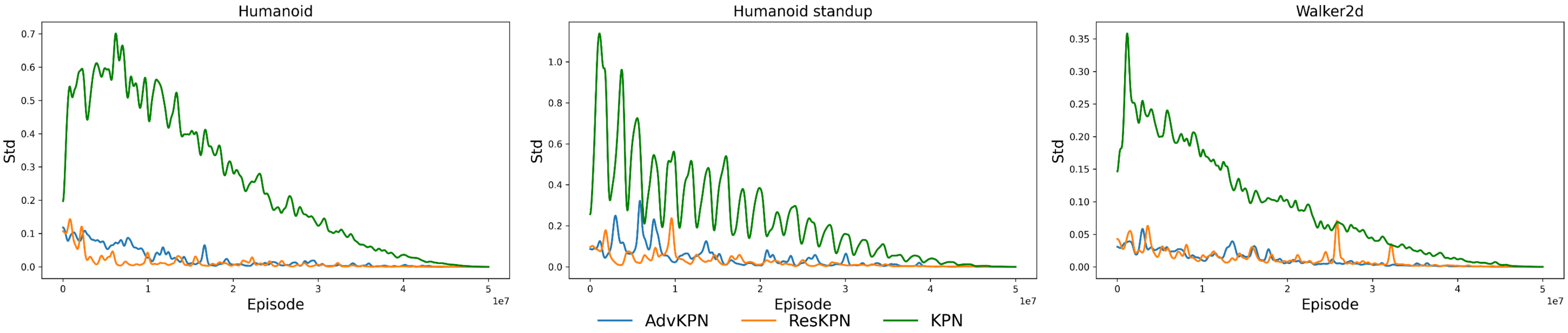




The representation learning increase the robustness of RKHS policy by adjusting the distribution, while its dynamically representation contribute to the overall performance greatly.

# Effectiveness Analysis

TBSI 清华-伯克利深圳学院 Tsinghua-Berkeley Shenzhen Institute

**For the variance reduction:** We testify the variance of gradient in the training




The use of advantage stabilize the training greatly, and the residual layer further reduce the variance in training.

# Conclusion

Tsinghua-Berkeley Shenzhen Institute

**Potential Follow-up Studies:**

**(a)** The use of RKHS serves as an additional module in policy gradient algorithms, where it can be used in MARL problem.
**(b)** The minibatch method can also be used in RKHS gradient method.

**Limitations:** Computational cost (in minutes):

| Environment | KPN | Gaussian ResKPN | PPO | DPO | Laplacian ResKPN | Linear ResKPN | Sigmoid ResKPN |
|---|---|---|---|---|---|---|---|
| Half Cheetah | 13.09 | 13.18 | 3.95 | 4.68 | 14.21 | 6.08 | 6.24 |
| Humanoid Standup | 13.70 | 13.77 | 2.08 | 2.14 | 12.56 | 7.37 | 6.61 |
| Inverted Pendulum | 10.29 | 10.31 | 3.50 | 3.86 | 11.22 | 2.60 | 2.63 |
| Walker2d | 11.00 | 11.06 | 3.69 | 4.12 | 10.59 | 3.55 | 3.69 |
| Hopper | 10.69 | 10.80 | 1.54 | 1.55 | 13.56 | 3.30 | 3.34 |
| Humanoid | 13.07 | 13.23 | 2.13 | 2.28 | 15.43 | 5.96 | 6.12 |

# Thanks for your listening