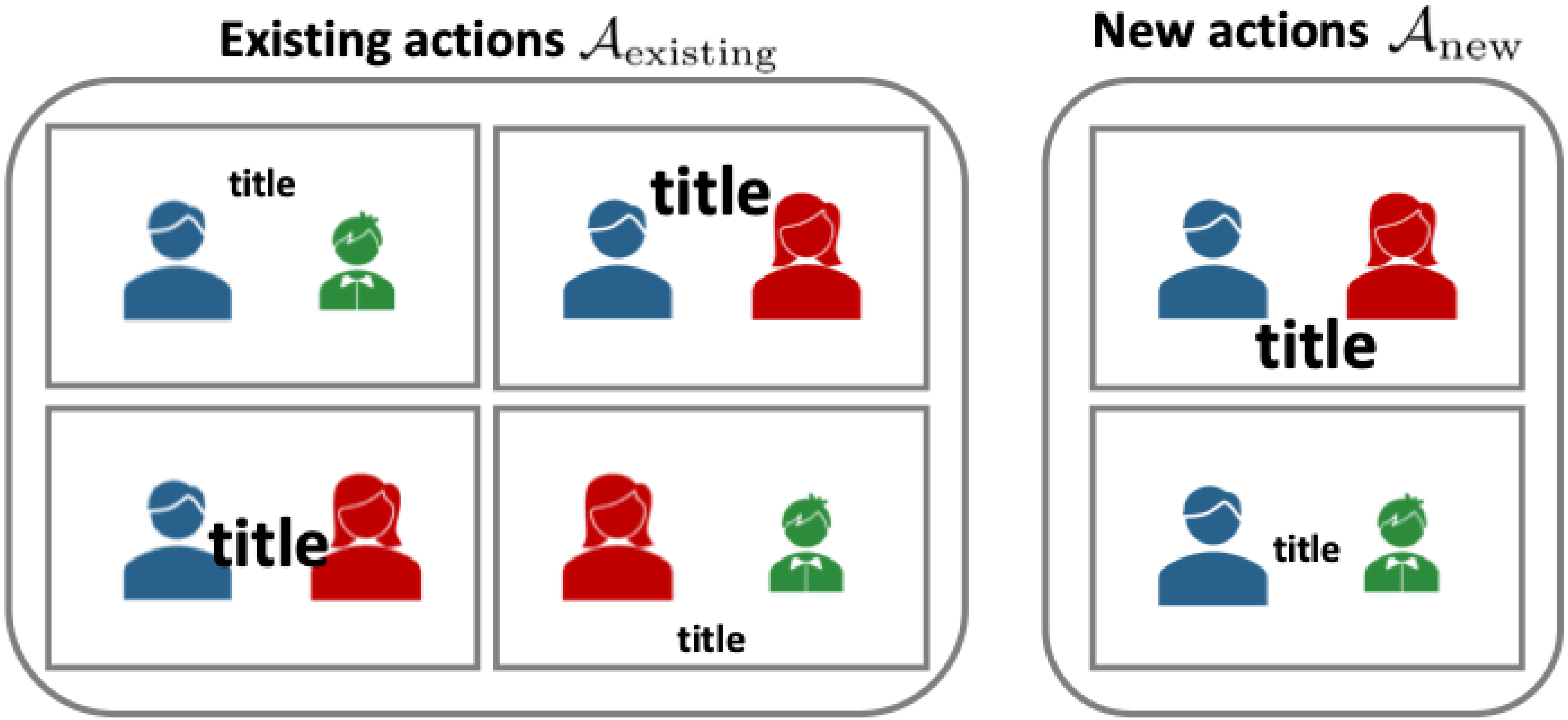
Offline Contextual Bandits in the Presence of New Actions

Ren Kishimoto, **Tatsuhiro Shimizu**, Kazuki Kawamura, Takanori Muroi, Yusuke Narita, Yuki Sasamoto, Kei Tateno, Takuma Udagawa, Yuta Saito

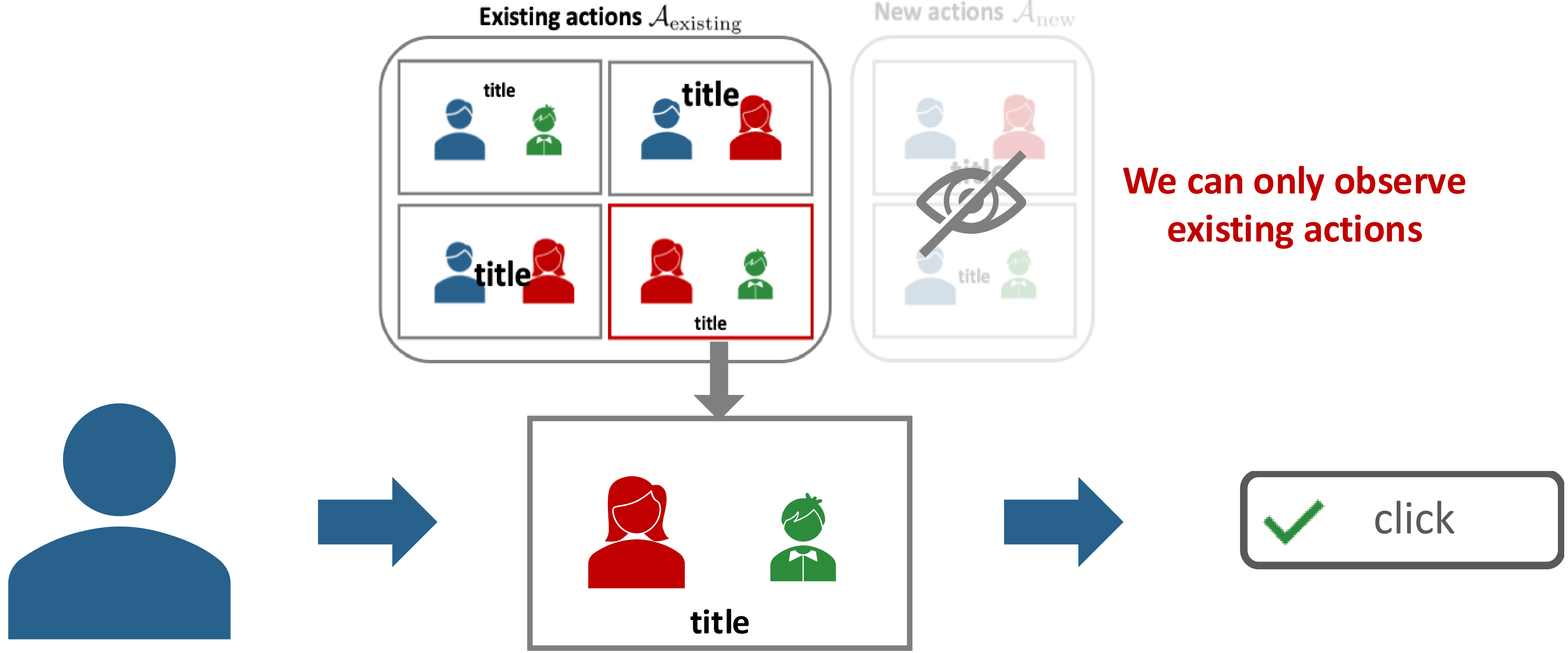
August 3-7, 2025
KDD2025

Motivation

Q: How can we effectively learn a policy where there exist new actions?



Data Generating Process in Contextual Bandits



$x \sim p(x)$

Context
(e.g., user)

$a \sim \pi_0(a|x)$

Action
(e.g., thumbnail)

$r \sim p(r|x, a)$

Reward
(e.g., conversion)

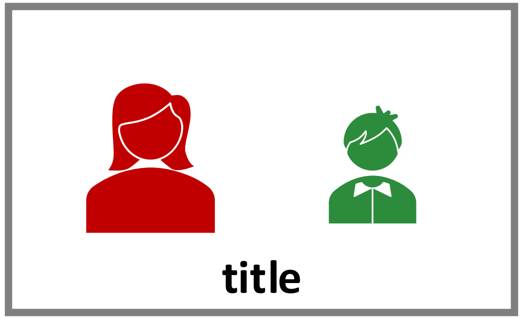
Logged Bandit Data in Contextual Bandits

Logged Bandit Data

$$\mathcal{D} := \{(x_i, a_i, r_i)\}_{i=1}^{n} \sim \prod_{i=1}^{n} p(x_i)\underbrace{\pi_0(a_i|x_i)}_{\text{behavior/logging policy}}p(r_i|x_i, a_i)$$

$x \sim p(x)$

Context



$a \sim \pi_0(a|x)$

Action

$r \sim p(r|x, a)$

Reward

Problem of Off-policy Learning (OPL)

Goal of OPL: Learn a parameterized policy which maximizes the policy value

Goal of OPL

$$\theta^* = \mathrm{argmax}_{\theta \in \Theta} V(\underbrace{\pi_\theta}_{\text{parameterized policy}})$$

The performance metric of OPL is the expected reward under a policy

Policy Value

$$V(\pi) := \mathbb{E}_{p(x)\pi(a|x)}[\underbrace{q(x,a)}_{\substack{\text{expected reward} \\ \text{given context and action}}}]$$

Existing Method: Policy-based Method

Policy-based methods use the policy gradient to iteratively update the parameter

Iterative Parameter Update

$$\theta_{t+1} \leftarrow \theta_t + \eta \underbrace{\nabla_\theta V(\pi_\theta)}_{\text{policy gradient}}$$

Since we cannot access the true policy gradient, we need to estimate it

<u>Inverse Propensity Scoring (IPS)</u>

$$\nabla_\theta \widehat{V}_{\mathrm{IPS}}(\pi_\theta; \mathcal{D}) := \frac{1}{n} \sum_{i=1}^{n} \frac{\pi_\theta(a_i|x_i)}{\pi_0(a_i|x_i)} r_i \nabla_\theta \log \pi_\theta(a_i|x_i)$$

<u>Doubly Robust (DR)</u>

$$\nabla_\theta \widehat{V}_{\mathrm{DR}}(\pi_\theta; \mathcal{D}) := \frac{1}{n} \sum_{i=1}^{n} \left\{ \frac{\pi_\theta(a_i|x_i)}{\pi_0(a_i|x_i)} (r_i - \hat{q}(x_i, a_i)) \nabla_\theta \log \pi_\theta(a_i|x_i) + \sum_{a \in \mathcal{A}} \pi_\theta(a|x_i) \hat{q}(x_i, a) \nabla_\theta \log \pi_\theta(a|x_i) \right\}$$

Existing Method: Properties of IPS and DR

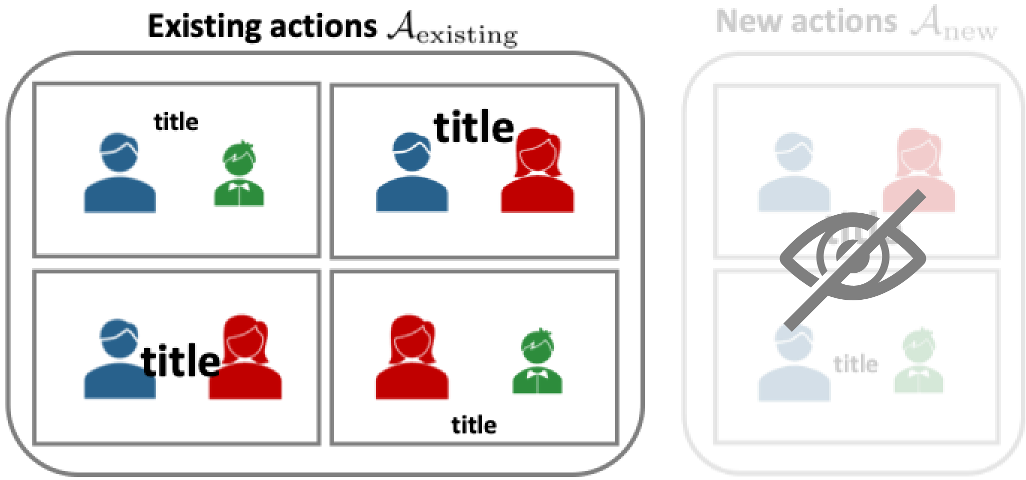
IPS and DR are unbiased under **full support**

Full Support

$$\pi_0(a|x) > 0$$

$$\forall x \in \mathcal{X}, \forall a \in \mathcal{A}$$

However, IPS and DR do not select a new action at all



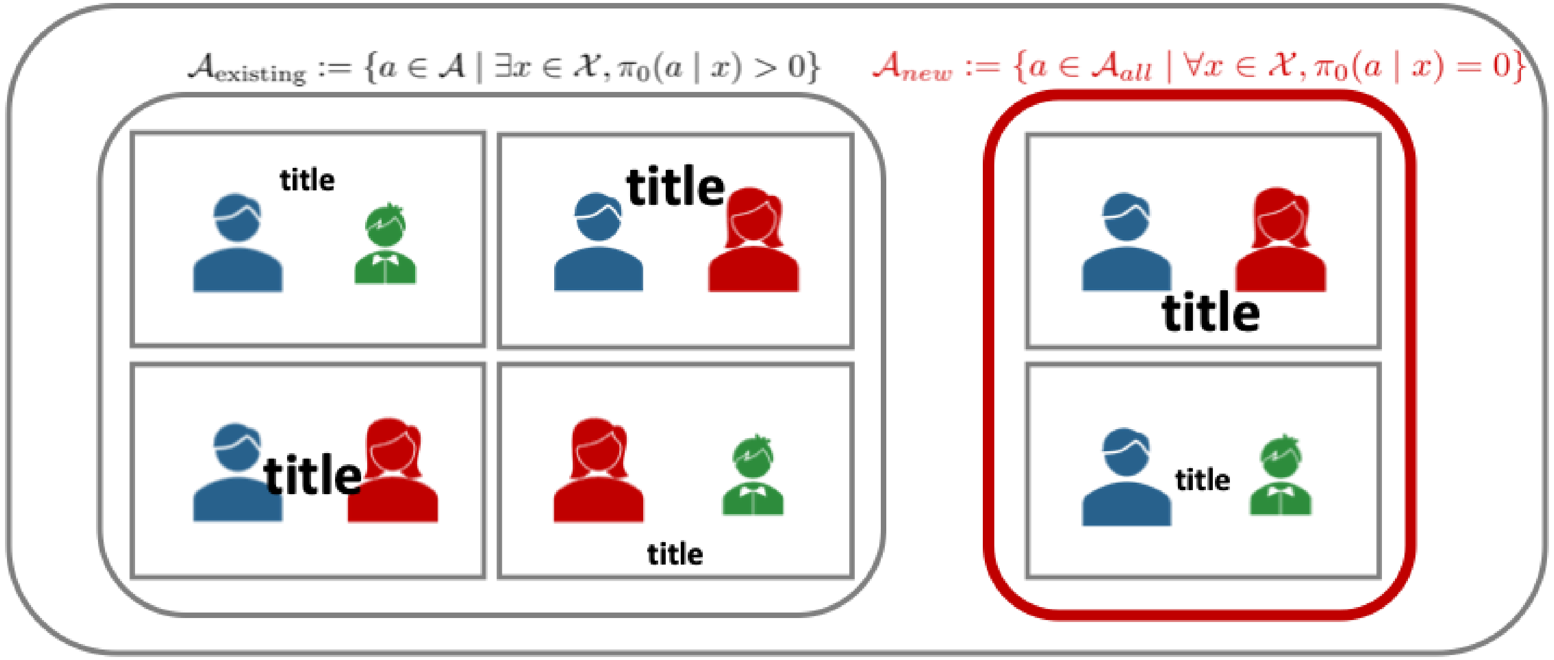
Definition of the Set of New Actions

We represent action a as **d-dimensional action features**

$$f(a) = (f_1(a), \ldots, f_l(a), \ldots, f_d(a))$$

We define **new actions** as the combination of the action features whose probability is 0 for any context

All actions $\mathcal{A} := \{a \mid a = (f_1, \ldots, f_d), \forall f_1 \in \mathcal{F}_1, \ldots, \forall f_d \in \mathcal{F}_d\}$



Action Features

1. Character type
   - male, female, child
2. Title position
   - top, center, bottom
3. Title size
   - small, large

Key Idea 1: Relaxation of Full Support

Independent support considers the support for each dimension of the action feature

Independent Support

$$\pi_0(\underline{f_l}|x) > 0$$

Support for each dimension of action feature

$$\forall x \in \mathcal{X}, \forall l \in [1, ..., d], \forall f_l \in \mathcal{F}_l$$

where $\pi_0(f_l|x) := \sum_{a \in \mathcal{A}: f_l(a) = f_l} \pi_0(a|x)$ is the marginal probability of observing f_l under π_0

Independent support is a **weaker assumption** than full support

The Pseudoinverse (PI) Estimator

Pseudoinverse estimator is based on the <u>independent support</u>

Pseudoinverse (PI)

$$\nabla_\theta \widehat{V}_{\mathrm{PI}}(\pi_\theta; \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{a \in A} \pi_\theta(a|x_i) \nabla_\theta \log \pi_\theta(a|x_i) \underline{\mathbb{I}_{f_l(a)}^T} \right) \Gamma^\dagger_{\pi_0, x_i} \mathbb{I}_{f_l(a)_i} r_i$$

Flattened vector representing the one-hot encoding of each action feature

where $\Gamma_{\pi_0,x} := \mathbb{E}_{\pi_0(a|x)}[\mathbb{I}_{f_l(a)} \mathbb{I}_{f_l(a)}^T | x]$ and

$M^\dagger$ denotes the Moore-Penrose pseudoinverse of matrix M

PI can learn a new action thanks to the relaxation of the support condition

Property of PI

PI is unbiased under **independent support** and **linearity**

Linearity

$$q(x,a) = \sum_{l=1}^{d} \underbrace{q_l(x, f_l(a))}_{\substack{\text{Latent value function} \\ \text{for each dimension of action feature}}} = \mathbb{I}_{f_l(a)}^T \underbrace{\phi_{x,l}}_{\text{Intrinsic reward vector}}$$

However, linearity is rarely satisfied in practice

Key Idea 2: Relaxation of Linearity

Local linearity allows the interaction of the first s dimensions of the action features

Local Linearity

$$q(x,a) = \underbrace{\sum_{l=1}^{d} \underbrace{q_l(x, f_l(a))}_{\text{Latent value for each dimension}} + \underbrace{q(x, f_{1:s}(a))}_{\text{Interaction effect of the first } s \text{ dimensions}}}_{\mathbb{I}^T_{f_l(a)}\phi_{x,l} + \underbrace{\mathbb{I}^T_{f_{1:s}(a)}}_{\text{Vector with binary values representing the first } s \text{ dimensions}}\phi_{x,1:s}} = \underbrace{\mathbb{I}_a^T}_{\text{Overall action indicator}} \phi_x .$$

where $\mathbb{I}_a := \mathrm{concat}[\mathbb{I}_{f_l}, \mathbb{I}_{f_{1:s}}]$ and $\phi_x := \mathrm{concat}[\phi_{x,l} \in \mathbb{R}^{dm}, \phi_{x,1:s} \in \mathbb{R}^{m^s}]$

Local linearity is a **weaker assumption** than linearity

The Local Combination Pseudoinverse (LCPI) Estimator

LCPI allows the interaction effects of first s dimensions of action features

Local Combination Pseudoinverse (LCPI)

$$\nabla_\theta \widehat{V}_{\mathrm{LCPI}}(\pi_\theta; \mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{a\in A}\pi_\theta(a|x_i)\nabla_\theta \log \pi_\theta(a|x_i)\mathbb{I}_a^T\right)\Gamma_{\pi_0,x_i}^{\dagger}\mathbb{I}_{a_i} r_i$$

where $\Gamma_{\pi_0,x} := \mathbb{E}_{\pi_0(a|x)}[\mathbb{I}_a\mathbb{I}_a^T|x]$ and $\mathbb{I}_a := \mathrm{concat}[\mathbb{I}_{f_l}, \underline{\mathbb{I}_{f_{1:s}}}]$

Allow the Interaction effect
of the first s dimensions

PI is the special case of LCPI where $s = 1$

Property of LCPI

LCPI is unbiased under **local linearity** and **local combination support**

Local Combination Support

$$\pi_0(\underline{f_{1:s}}|x) > 0, \forall x \in \mathcal{X}, \forall f_{1:s} \in \prod_{j=1}^{s} \mathcal{F}_j$$

Support for the first s dimensions of action feature

$$\pi_0(\underline{f_l}|x) > 0, \forall l \in \{s+1, \ldots, d\}, \forall x \in \mathcal{X}, \forall f_l \in F_l$$

Independent support for the rest of the dimensions

Thus, LCPI can effectively select a new action under the mild assumptions

Key Idea 3: Balance Tradeoff between Policy Value and New Actions

Combining LCPI and DR will yield the high policy value and effective new actions

OPL Method	Overall Learned Policy Value	Ability to Learn New Actions
RegressionBased (a)	Medium	No
PolicyBased (IPS)	Medium	No
PolicyBased (DR)	High	No
PolicyBased (PI)	Medium	Yes
PolicyBased (LCPI)	Medium-High	Yes

Combine two great properties

The Policy Optimization for New Actions (PONA) Algorithm

PONA takes the **weighted average of LCPI and DR**

Policy Optimization for New Actions (PONA)

$$\nabla_\theta \hat{V}_{\mathrm{PONA}}(\pi_\theta; \kappa, \mathcal{D}) = \underline{\kappa} \cdot \nabla_\theta \hat{V}_{\mathrm{LCPI}}(\pi_\theta; \mathcal{D}) + \underline{(1-\kappa)} \cdot \nabla_\theta \hat{V}_{\mathrm{DR}}(\pi_\theta; \mathcal{D})$$

***κ* balances the policy value and learning new actions**

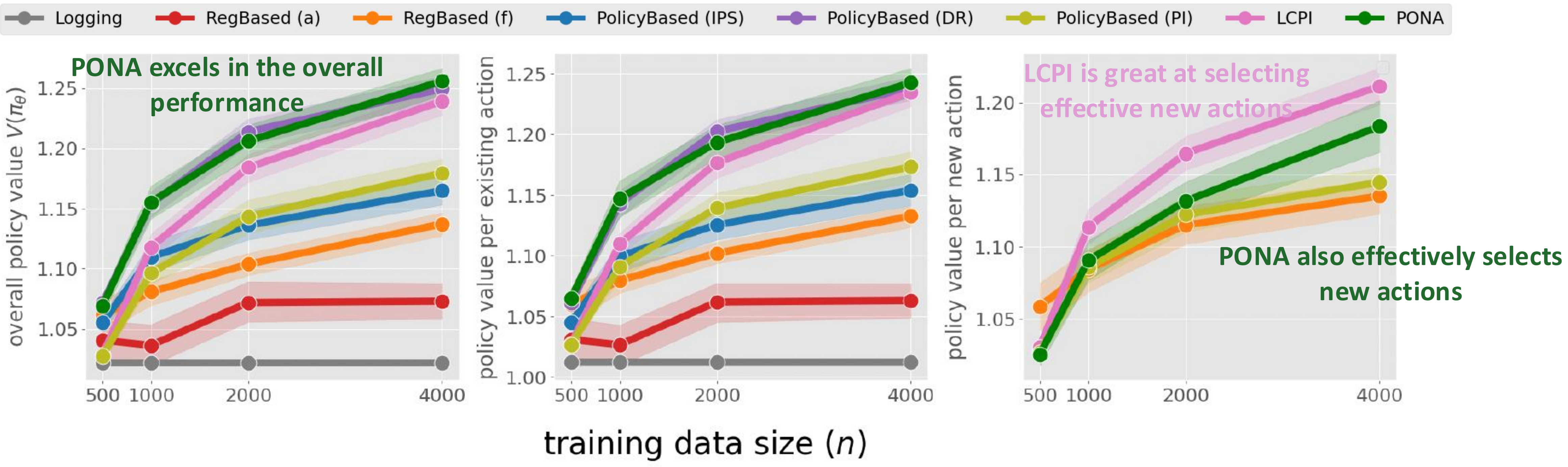
We can impose the following constraints for hyperparameter tuning

Constraints on the percentage of new actions

$$\max_{\kappa} \ \hat{V}(\pi_{\theta,\kappa}; \mathcal{D}) \quad \text{s.t. } \underbrace{\rho_L}_{\text{Lower limit}} \leq \underbrace{\mathbb{E}_{p(x)}\left[\sum_{a \in \mathcal{A}_{new}} \pi_{\theta,\kappa}(a|x)\right]}_{\text{Percentage of new actions}} \leq \underbrace{\rho_U}_{\text{Upper limit}}$$

Synthetic Data Experiment with Varying Training Data Size

- PONA **effectively learns new actions** while achieving **the highest policy value**, tying with DR
- LCPI excels in the **selection of effective new actions**



Summary

- **Existing OPL** methods can effectively select existing actions but **cannot explore new actions at all**

- **PI** can **select a new action** due to its basis on independent support on action features

- **LCPI** further **improves the effectiveness of new actions** by relaxing linearity

- Finally, **PONA** balances the tradeoff of the **overall policy value** optimization and learning **new actions** via the hyperparameter

Appendix

Existing Method: Regression-based Method

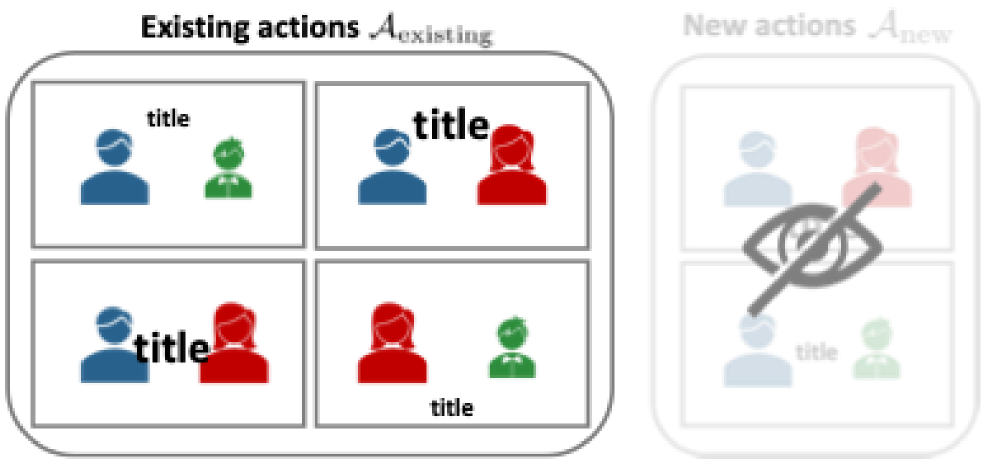
Regression-based methods learn a policy via the estimation of the q-function $\hat{q}_\theta(x, a)$

Typical Regression-based Methods

$$\pi_\theta(a|x) = \frac{\exp(\hat{q}_\theta(x, a)/\tau)}{\sum_{a'} \exp(\hat{q}_\theta(x, a')/\tau)}$$

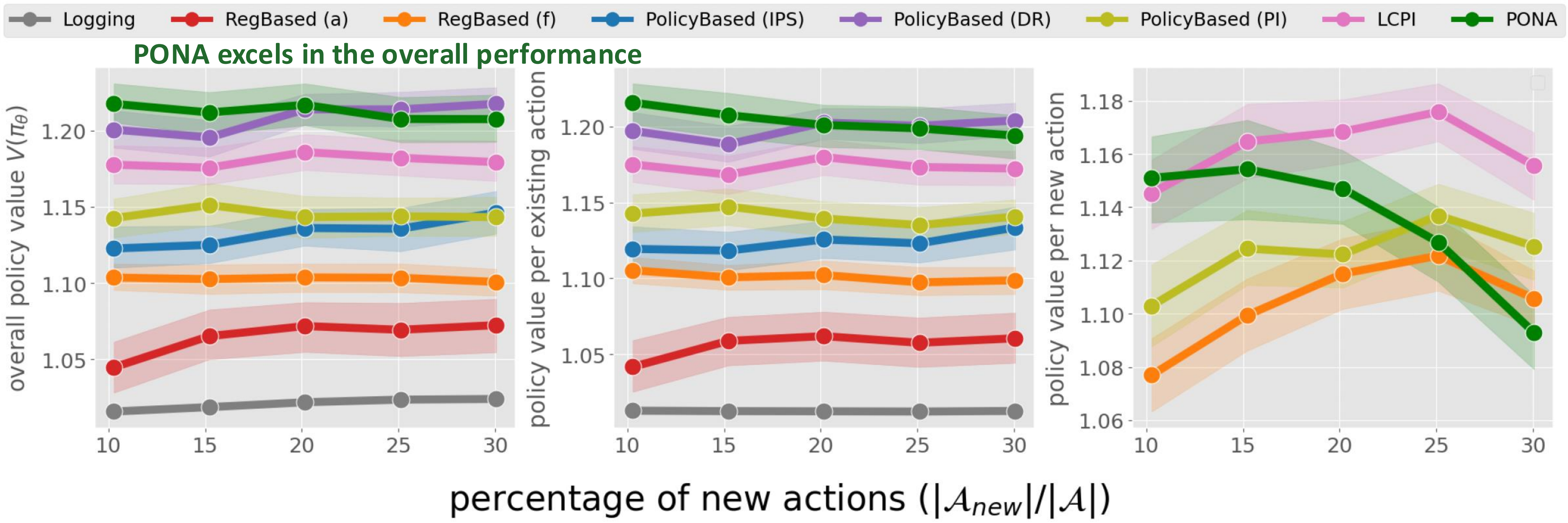
where $\tau > 0$ is the temperature parameter

However, it cannot select a new action at all



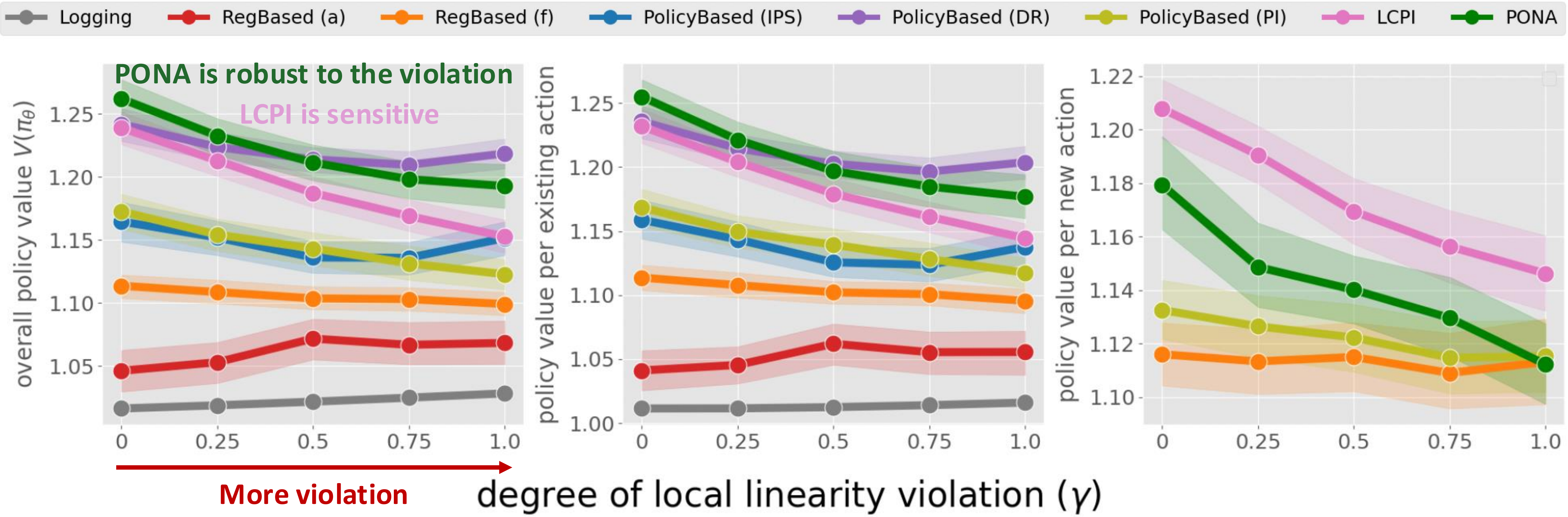
Synthetic Data Experiment with Varying Number of New Actions

- PONA learns new actions while **achieving the higher or same performance compared to PolicyBased (DR)** <u>even when there are many new actions</u>
- LCPI achieves **higher policy values in each metric** compared to PI due to the <u>relaxation of reward assumption</u>



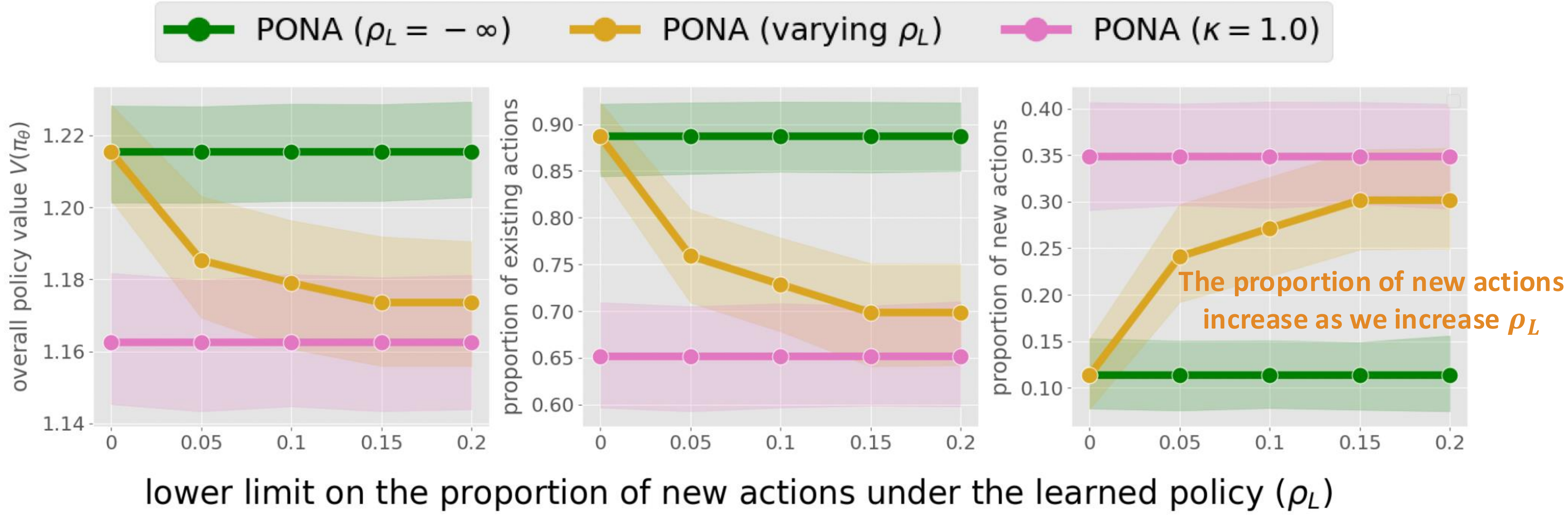
Synthetic Data Experiment with Varying Degree of Local Linearity

- PONA is **more robust to the violation** of the local linearity
- LCPI is sensitive to the violation of the local linearity
- Existing methods are not affected by the violation of local linearity



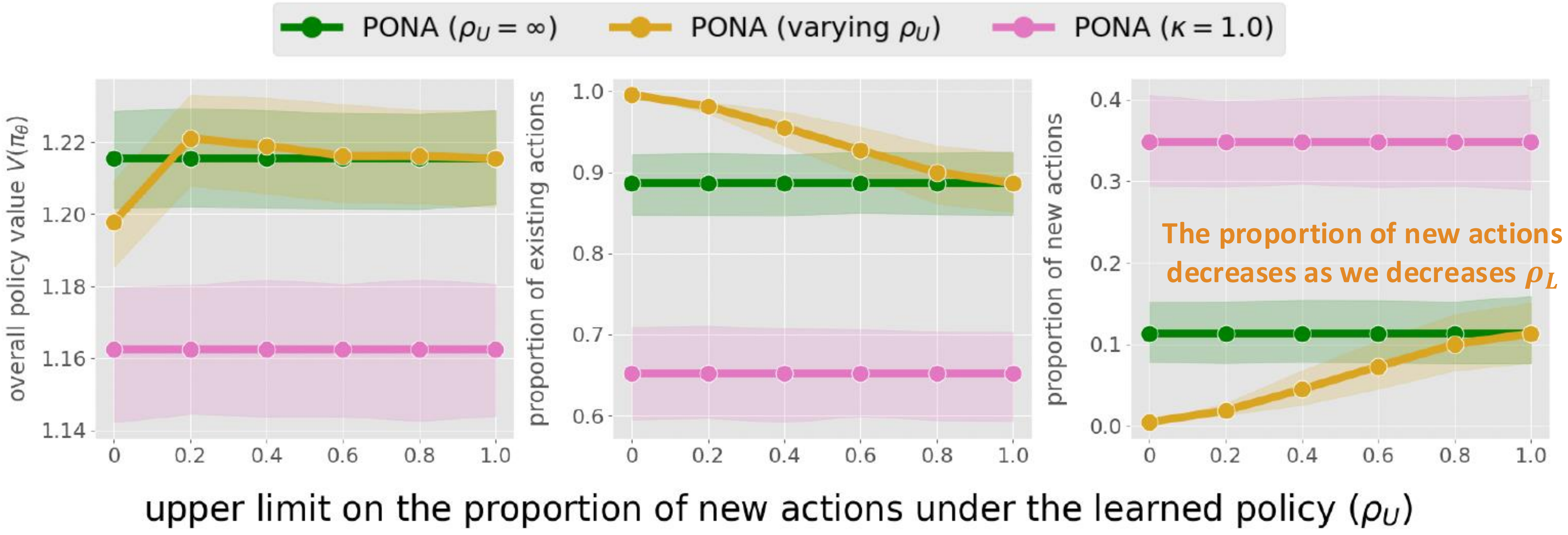
Synthetic Data Experiment with Varying Lower Limit

- The proportion of the new actions increases as we increase the lower limit
- The hyperparameter tuning of κ can **effectively control the proportion of new actions**



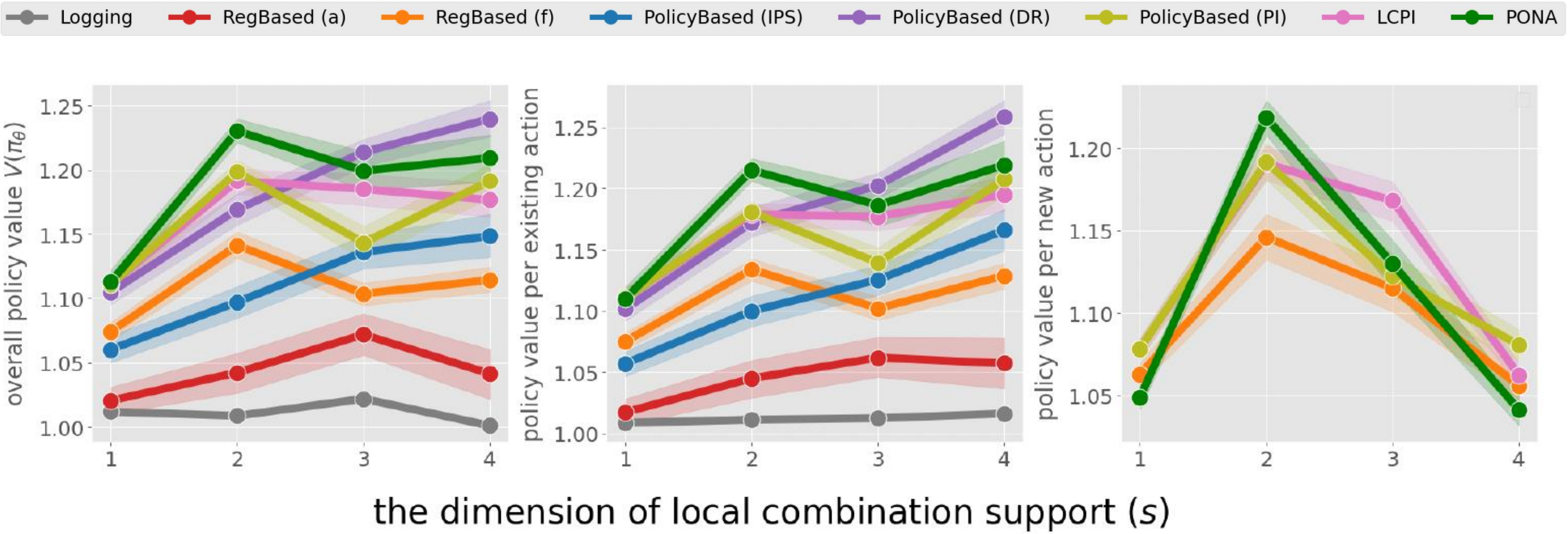
Synthetic Data Experiment with Varying Upper Limit

- The proportion of the new actions decreases as we decrease the upper limit
- The hyperparameter tuning of κ can **effectively control the proportion of new actions**



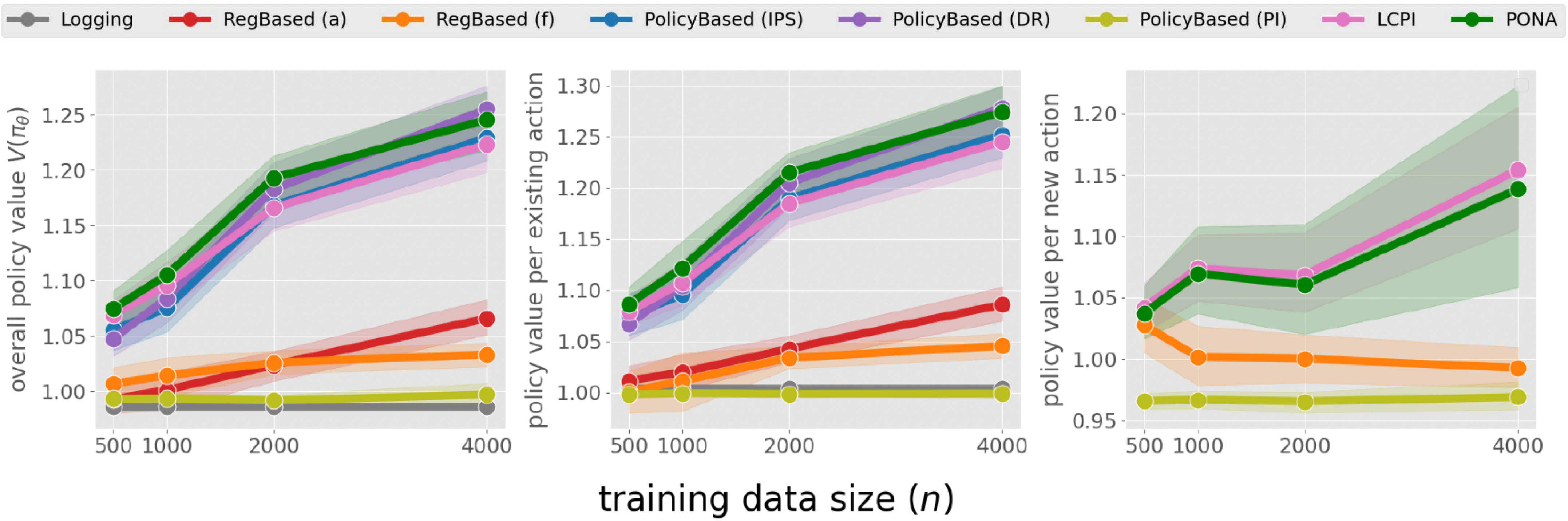
Synthetic Data Experiment with Varying Dimension of Local Combination Support

- PONA and LCPI can learns new actions while **achieving the comparable performance with** DR under various dimension of local combination support
- LCPI is the same as PI when $s = 1$



Real-world Data Experiment with Varying Training Data Size

- PONA **effectively learns new actions** while achieving **the highest policy value**, tying with DR
- DR does not choose new actions at all



Real-world Data Experiment with Percentage of New Actions

- PONA learns new actions while **achieving the higher or same performance compared to** DR even when there are many new actions
- LCPI achieves **higher policy values in each metric** compared to PI due to the relaxation of reward assumption