Off-Policy Evaluation for Large Action Spaces via Policy Convolution

Noveen Sachdeva*
nosachde@ucsd.edu
University of California, San Diego
La Jolla, CA, USA

Lequn Wang
lequnw@netflix.com
Netflix
Los Gatos, CA, USA

Dawen Liang
dliang@netflix.com
Netflix
Los Gatos, CA, USA

Nathan Kallus
kallus@cornell.edu
Netflix & Cornell University
Los Gatos, CA, USA

Julian McAuley
jmcauley@ucsd.edu
University of California, San Diego
La Jolla, CA, USA

ABSTRACT

Developing accurate off-policy estimators is crucial for both evaluating and optimizing for new policies. The main challenge in off-policy estimation is the distribution shift between the logging policy that generates data and the target policy that we aim to evaluate. Typically, techniques for correcting distribution shift involve some form of importance sampling. This approach results in unbiased value estimation but often comes with the trade-off of high variance. Furthermore, importance sampling relies on the common support assumption, which becomes impractical when the action space is large. To address these challenges, we introduce the Policy Convolution (PC) family of estimators for the contextual bandit setting. These methods leverage latent structure within actions—made available through action embeddings—to strategically *convolve* the logging and target policies. This convolution introduces a unique bias-variance trade-off, that can be controlled via the *amount of convolution*. Our experiments on synthetic and benchmark datasets demonstrate remarkable mean squared error (MSE) improvements when using PC, especially when either the action space or policy mismatch becomes large, with gains of up to $5-6$ orders of magnitude over existing estimators.

CCS CONCEPTS

• **Information systems** → **Evaluation of retrieval results**.

KEYWORDS

off-policy evaluation; inverse propensity score

ACM Reference Format:
Noveen Sachdeva, Lequn Wang, Dawen Liang, Nathan Kallus, and Julian McAuley. 2024. Off-Policy Evaluation for Large Action Spaces via Policy Convolution. In *Proceedings of the ACM Web Conference 2024 (WWW '24), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 36 pages. https://doi.org/10.1145/3589334.3645501

*Work done during internship at Netflix.

WWW '24, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0171-9/24/05
https://doi.org/10.1145/3589334.3645501

1 INTRODUCTION

Off-policy estimation (OPE) is a fundamental problem in reinforcement learning and decision making under uncertainty. It involves estimating the expected value of a target policy, given access to only an offline dataset logged by deploying a different policy, often referred as the logging policy (see [43] for a comprehensive survey). This decoupling between data collection and policy evaluation is crucial in many real-world applications, as it allows for the assessment of new policies using historical data without having to deploy them in the environment, which can be costly and/or risky. In this paper, we focus on OPE for the one-step contextual bandit setting, *i.e.*, we perform decision making with only an observed context that is assumed to be independently sampled (*e.g.*, a user coming to a website), and do not consider any recurrent dependencies in the context transitions as is the case in the general formulation of reinforcement learning.

OPE, in its most general setting, can be a very challenging problem due to its inherently counterfactual nature, as we observe the reward for only those actions taken by the logging policy, while we aim to evaluate *any* target policy. For example, consider a scenario where the logging policy in a movie recommendation platform, for a given segment of users, rarely recommends romantic movies. This can often happen when we think a user will not like certain type of movies. On the other hand, a target policy—whose value we aim to estimate—due to numerous potential reasons, now chooses to recommend romantic movies for the same user segment. This distribution-shift can lead to irrecoverable bias in our estimates [26], making it difficult to accurately evaluate a target policy or learn a better one, which typically involves optimizing over the value estimates [12, 37].

Typical off-policy estimators utilize Importance Sampling (IS) to correct for the policy mismatch between the target and logging policies [4, 10, 14, 21, 27, 30, 34, 36, 38, 42, 46], leading to unbiased value estimation, at the cost of high variance. The variance problem caused by IS is exacerbated if the target and logging policies exhibit significant divergence, and even more so if the action space is large. Notably, large action spaces frequently occur in practical OPE scenarios, *e.g.*, recommender systems which can have millions of items (actions) [25, 47], extreme classification [19, 22, 31], discretized continuous action-spaces [40], *etc.*

To address the aforementioned limitations of IS, we propose the Policy Convolution (PC) family of estimators. PC strategically

convolves the logging and target policies by exploiting the inherent latent-structure amongst actions—available through action-embeddings—to make importance sampling operate in a more favorable bias-variance trade-off region. Such structure can occur naturally in different forms like action meta-data (text, images, *etc.*), action hierarchies, categories, *etc.* Or they can be estimated using domain-specific representation learning techniques [2]. Notably, the utilization of additional action-structure has also been studied in the online multi-armed bandit literature (Lipschitz bandits) [15, 32, 33], albeit in the context of regret minimization.

To be more specific, the PC framework for OPE consists of two components: (1) conventional IS-based value estimation; and (2) convolving *both* the target and logging policies using action-action similarity. PC allows full freedom over the choice of the *backbone* IS *estimator*, the convolution function, as well as the *amount of convolution* to conduct on the target and logging policies respectively.

To better understand the practical effectiveness of PC, we compare its performance with various off-policy estimators on synthetic and real-world benchmark datasets, simulating a variety of off-policy scenarios. Our results demonstrate that PC can effectively balance the bias-variance trade-off posited by policy convolutions, leading to up to $5-6$ orders of magnitude better off-policy evaluation in terms of mean squared error (MSE), particularly when the action space is large or the policy mismatch is high. To summarize, *in this paper*, we make the following contributions:

- Introduce the Policy Convolution (PC) family of off-policy estimators that posit a novel bias-variance trade-off controlled by the *amount of convolution* on the logging and target policies.
- Propose four different convolution functions for the PC framework, where each convolution function is accompanied with its unique set of inductive biases, thereby leading to distinct performance comparisons.
- Conduct empirical analyses on synthetic and real-world benchmark datasets that demonstrate the superiority of PC over a variety of off-policy estimators, especially when the action space or policy mismatch becomes large.

2 PRELIMINARIES

2.1 OPE in Contextual Bandits

We study OPE in the standard stochastic contextual bandits setting with a context space $\mathcal{X}$ and afi nite action space $\mathcal{A}$. In each round i, the agent observes a context $x_i \in \mathcal{X}$, takes an action $a_i \in \mathcal{A}$, and observes a reward $r_i \in [0,1]$. The context x_i is drawn from some unknown distribution $p(x)$. The action a_i follows some policy $\pi(\cdot|x_i)$, and the reward r_i is draw from an unknown distribution $p(r|x_i,a_i)$ with expected value $\delta(a,x) \triangleq \mathbb{E}_{r\sim p(r|a,x)}[r]$. The value of a policy is its expected reward

$$V(\pi) \triangleq \mathop{\mathbb{E}}_{x\sim p(x)}\left[\mathop{\mathbb{E}}_{a\sim\pi(\cdot|x)}[\delta(a,x)]\right].$$

In OPE, given a target policy π, we aim to estimate its value $V(\pi)$ using some bandit feedback data $\mathcal{D} \triangleq \{(x_i,a_i,r_i)\}_{i=1}^n$ collected by deploying a different policy μ. We call μ the logging policy and assume it is known.

2.2 Conventional OPE Estimators

We now briefly discuss a few prominent OPE estimators, which will also be used to instantiate our proposed Policy Convolution (PC) estimator discussed in Section 3.

2.2.1 Direct Method (DM). Taking a model-based approach, DM leverages a reward-model to estimate the value of the target policy. Formally, given a suitable $\hat{\delta}: \mathcal{A}\times\mathcal{X}\mapsto \mathbb{R}$, the estimator is defined as follows:

$$\hat{V}_{\mathtt{DM}}(\pi) \triangleq \mathop{\mathbb{E}}_{(x,\cdot,\cdot)\sim\mathcal{D}}\left[\sum_{a\in\mathcal{A}}\pi(a|x)\cdot\hat{\delta}(a,x)\right],$$

where the outer expectation is over thefi nite set of logged bandit feedback data $\mathcal{D}$. Notably, the variance of $\hat{V}_{\mathtt{DM}}(\cdot)$ is often quite low, since $\hat{\delta}$ is typically bounded. However, it can suffer from a large bias problem due to model misspecification [5].

2.2.2 Inverse Propensity Scoring (IPS). IPS [10] estimator uses Monte-Carlo approximation and importance sampling to account for the policy-mismatch between π and μ as follows:

$$\hat{V}_{\mathtt{IPS}}(\pi) \triangleq \mathop{\mathbb{E}}_{(x,a,r)\sim\mathcal{D}}\left[\frac{\pi(a|x)}{\mu(a|x)}\cdot r\right].$$

The IPS estimator is unbiased under the following two assumptions which we assume throughout the paper unless otherwise specified:

Assumption2.1. **(Unconfoundedness)** *The action selection procedure is independent of all potential outcomes given the context, i.e.,* $\forall x\in\mathcal{X}, \pi\in\Pi, a\sim\pi(\cdot|x):\ \{\delta(a',x)\}_{a'\in\mathcal{A}} \perp a \mid x$.

Assumption2.2. **(Common Support)** *The target policy* π *shares common support with the logging policy* μ, $\forall x\in\mathcal{X}, a\in\mathcal{A}$: $\pi(a|x) > 0 \implies \mu(a|x) > 0$.

However, IPS estimator can suffer from a large variance problem, since the importance weights $\pi(a|x)/\mu(a|x)$ can be unbounded and huge. Several estimators are proposed to reduce the variance of the IPS estimator.

2.2.3 Self-normalized Inverse Propensity Scoring (SNIPS). Built on the observation that the expected propensity weight in IPS equals 1 , SNIPS [38] uses the empirical average of the propensity weights as a control variate for IPS as follows:

$$\hat{V}_{\mathtt{SNIPS}}(\pi) \triangleq \mathop{\mathbb{E}}_{(x,a,r)\sim\mathcal{D}}\left[\frac{\pi(a|x)}{\rho\cdot\mu(a|x)}\cdot r\right] \text{ s.t. } \rho \triangleq \mathop{\mathbb{E}}_{(x,a,\cdot)\sim\mathcal{D}}\left[\frac{\pi(a|x)}{\mu(a|x)}\right].$$

SNIPS typically enjoys smaller variance at the cost of a slight added bias in comparison to IPS, especially when the variance of the propensity weight is large[8]. Further, $\hat{V}_{\mathtt{SNIPS}}(\pi)$ is a strongly consistent estimator of $V(\pi)$ by the law of large numbers.

2.2.4 Doubly Robust (DR). DR combines the benefits of unbiased estimation in IPS and the low-variance, model-based estimation in DM:

$$\hat{V}_{\mathtt{DR}}(\pi) \triangleq \mathop{\mathbb{E}}_{(x,a,r)\sim\mathcal{D}}\left[\frac{\pi(a|x)}{\mu(a|x)}\cdot(r-\hat{\delta}(a,x))+\Delta(\pi,x)\right]$$
$$\text{s.t. } \Delta(\pi,x) \triangleq \sum_{a'\in\mathcal{A}}\pi(a'|x)\cdot\hat{\delta}(a',x),$$

where $\hat{\delta}$ is the same reward-model as used in DM (Section 2.2.1). Intuitively, DR uses the reward-model as a baseline, and performs

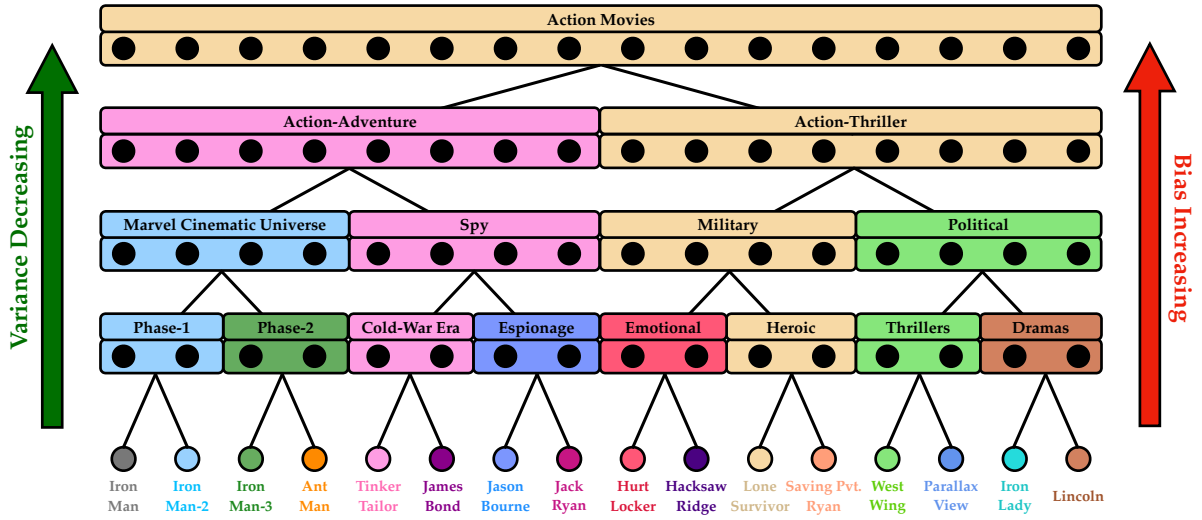

Figure 1: Intuition for the PC framework demonstrated using a hierarchical action tree, where similar actions (movies in this example) are recursively agglomerated together. The last level (leaf nodes) represents the complete action space, and the higher levels consist of "meta-actions" that represent a group of individual actions. As we go higher, PC defines the convolved policy for a given action, as the mean probability of all actions inside its corresponding *meta-action*. Hence, we obtain the uniform policy at the topmost-level, and recover the original policy at the last-level.

importance sampling only on the error of the given reward-model. DR is unbiased and can be of smaller variance than IPS when the reward-model $\hat{\delta}$ is close to the true reward δ [4].

2.2.5 Self-normalized Doubly Robust (SNDR). Similar to the idea behind SNIPS (Section 2.2.3); SNDR [24, 42] performs the same control variate technique on the DR estimator (Section 2.2.4) as follows:

$$\hat{V}_{\mathsf{SNDR}}(\pi) \triangleq \mathop{\mathbb{E}}_{(x,a,r)\sim\mathcal{D}}\left[\frac{\pi(a|x)}{\rho\cdot\mu(a|x)}\cdot(r-\hat{\delta}(a,x))+\Delta(\pi,x)\right]$$

$$\text{s.t. } \rho \triangleq \mathop{\mathbb{E}}_{(x,a,\cdot)\sim\mathcal{D}}\left[\frac{\pi(a|x)}{\mu(a|x)}\right] \;;\; \Delta(\pi,x) \triangleq \sum_{a\in\mathcal{A}}\pi(a|x)\cdot\hat{\delta}(a,x).$$

Hence, SNDR encapsulates the ideas behind all the aforementioned estimators to conduct strongly consistent, low-variance policy value estimation that might perform well (in terms of MSE) in practice.

While effective to some extent, the importance sampling based estimators mentioned above can still suffer from large variance due to large importance sampling weights, especially when the action space is large. In particular, the variance of these importance sampling based estimators grows roughly linearly *w.r.t.* the maximum propensity weight in $\mathcal{D}$. And the maximum propensity weight can grow linearly *w.r.t.* the size of action space $\Omega(|\mathcal{A}|)$ [39], making these estimators undesirable for OPE for large action-space problems. Further, when Assumption 2.2 is violated, the variance of such importance sampling based estimators becomes unbounded, in addition to incurring a bias of $\mathbb{E}_x\left[\sum_{a\in\mathcal{U}(x)}\pi(a|x)\delta(a|x)\right]$, where $\mathcal{U}(x)$ is the set of actions where $\mu(\cdot|x)$ doesn't put any probability mass on (blind spots) [26].

To address the aforementioned problems of importance sampling based estimators, we introduce policy convolution that makes use of the latent structure within actions in the next section.

3 OPE VIA POLICY CONVOLUTION

In addition to the offline dataset $\mathcal{D}$, we further posit access to some embeddings $\mathcal{E}: \mathcal{A}\mapsto \mathbb{R}^d$ of the actions, which maps an action a to a d-dimensional embedding space $\mathcal{E}(a)\in\mathbb{R}^d$. Let $\mathcal{E}_{\mathcal{A}}\subset\mathbb{R}^d$ be the subspace spanned by $\mathcal{E}$. Ideally, the embedding should capture action-similarity information, *i.e.*, smaller distance in the embedding space should imply smaller difference in terms of expected reward for each context $x\in\mathcal{X}$. Notably, such action-embeddings are typically readily available in many industrial recommender systems, *e.g.*, via matrix factorization [16].

We are now ready to define our Policy Convolution framework for OPE. Taking IPS (Section 2.2.2) as a representative "backbone" estimator for PC, we define PC-IPS as follows:

$$\hat{V}_{\mathsf{PC-IPS}}(\pi) \triangleq \mathop{\mathbb{E}}_{(x,a,r)\sim\mathcal{D}}\left[\frac{(\pi(\cdot|x)*f_{\tau_1})(a)}{(\mu(\cdot|x)*f_{\tau_2})(a)}\cdot r\right]$$
$$= \mathop{\mathbb{E}}_{(x,a,r)\sim\mathcal{D}}\left[\frac{\sum_{a'}\pi(a'|x)\cdot f_{\tau_1}(\mathcal{E}(a),\mathcal{E}(a'))}{\sum_{a'}\mu(a'|x)\cdot f_{\tau_2}(\mathcal{E}(a),\mathcal{E}(a'))}\cdot r\right],$$

where '$*$' represents the convolution operator specified in the action-embedding domain, and $f_\tau:\mathbb{R}^d\times\mathbb{R}^d\mapsto\mathbb{R}$ is an action-action similarity (or convolution) function which has a parameter 'τ' to control the amount of convolution. Notably, PC is not limited to the IPS backbone estimator discussed hitherto, and we analogously define PC for other backbone estimators, namely, Self-Normalized IPS (SNIPS), Doubly Robust (DR), Self-Normalized DR (SNDR) discussed in Sections 2.2.3 to 2.2.5, and call such estimators PC-SNIPS, PC-DR, and PC-SNDR for convenience. We provide their exact specifications in Appendix A.

		a_1	a_2	a_3	a_4	$V^*(\cdot)$	IPS with $\lvert\mathcal{D}\rvert = 10$		
							MSE	Bias^2	Var
$\delta(\cdot, x_1)$		5	10	15	20	–	–	–	–
$\tau = 1$	$\pi(\cdot\|x_1)$	0.0	0.2	0.2	0.6	17	48.0	≈ 0	48.0
	$\mu(\cdot\|x_1)$	0.2	0.2	0.4	0.2	13			
$\tau = 2$	$(\pi * f_\tau)(\cdot\|x_1)$	0.1	0.1	0.4	0.4	15.5	13.4	4.6	8.8
	$(\mu * f_\tau)(\cdot\|x_1)$	0.2	0.2	0.3	0.3	13.5			
$\tau = 3$	$(\pi * f_\tau)(\cdot\|x_1)$	0.25	0.25	0.25	0.25	12.5	18.6	16	2.6
	$(\mu * f_\tau)(\cdot\|x_1)$	0.25	0.25	0.25	0.25	12.5			

Table 1: A toy example to intuit PC using the Tree similarity function, and constrained to have $\tau_1 = \tau_2$. Similar to Figure 1, in this toy example, the action tree is a 3-level complete binary tree, where the action partitioning is defined as $\{(a_1, a_2, a_3, a_4)\} \rightarrow \{(a_1, a_2), (a_3, a_4)\} \rightarrow \{(a_1), (a_2), (a_3), (a_4)\}$ from top to bottom.

We also illustrate the intuition behind policy convolutions in Figure 1, where PC *strategically* biases the logging and target policies toward the uniform policy, by leveraging the underlying action structure, in this case, specified via a hierarchical grouping of actions. As we will observe in Section 4.2, such convolutions in-turn lead to a new bias-variance trade-off, controlled by the amount of convolution (τ). We propose four suitable instantiations of the action-action convolution (or pooling) function, $f_\tau(\cdot, \cdot)$ for PC:

- **Kernel Smoothing.** Perhaps the most intuitive, we use the idea of multi-variate kernel smoothing [44] in the action-embedding space to derive our similarity function as:

$$f_\tau\left(\mathcal{E}(a), \mathcal{E}(a')\right) \triangleq \mathcal{K}\left(\mathcal{E}(a), \mathcal{E}(a')\right) = \frac{1}{\tau^d} \prod_{i=1}^{d} \mathbf{K}\left(\frac{\mathcal{E}(a)_i, \mathcal{E}(a')_i}{\tau}\right),$$

  where, $\mathbf{K}$ is a suitable kernel function (*e.g.*, Gaussian), and $\tau \in \mathbb{R}$ now corresponds to the bandwidth. It is worth noting that such a formulation can also be derived by viewing actions as continuous treatments, as defined by their embeddings [9, 14]. However, since an inverse mapping from $\mathbb{R}^d \mapsto \mathcal{A}$ does not exist in our discrete action problem, having treatments outside $\mathcal{E}_\mathcal{A}$ is meaningless.
- **Tree Smoothing.** In this setting, we use $\mathcal{E}$ to recursively partition the action-space (see Figure 1 for a depiction) into a tree-like structure of depth D, where each depth can be specified by $\mathcal{T}_d \triangleq \{\mathbf{a}_{d,1}, \mathbf{a}_{d,2}, \ldots, \mathbf{a}_{d,k}\}$, where $\mathbf{a}_{d,i}$ is a *meta-action* (set) comprising of singular actions, such that $\mathcal{A} \equiv \bigcup_{i=1}^{k} \mathbf{a}_{d,i}$, and $\mathbf{a}_{d,i} \cap \mathbf{a}_{d,j} = \Phi$ for all $i \neq j$ pairs. Notably, $\mathcal{T}_D$ (root node) consists of a single meta-action with all actions, and $\mathcal{T}_1 \equiv \mathcal{A}$ (last level) consists of each singular action. The similarity function is then defined as:

$$f_\tau\left(\mathcal{E}(a), \mathcal{E}(a')\right) \triangleq \frac{\mathbb{I}(a' \in \mathbf{a}_\tau(a))}{|\mathbf{a}_\tau(a)|},$$

  where, $\mathbb{I}(\cdot)$ represents the indicator function, τ signifies the depth of the action-tree to use, and $\mathbf{a}_\tau(a)$ represents the meta-action at depth τ corresponding to the action a.
- **Ball Smoothing.** In this setting, we define a binary similarity function based on afi xed-radius ball around the given action, as defined by $\mathcal{E}$ as follows:

$$f_\tau\left(\mathcal{E}(a), \mathcal{E}(a')\right) \triangleq \frac{\mathbb{I}\left(\|\mathcal{E}(a) - \mathcal{E}(a')\|_2^2 < \tau\right)}{\left|\left\{\, \|\mathcal{E}(a) - \mathcal{E}(a'')\|_2^2 < \tau \mid \forall a'' \in \mathcal{A} \,\right\}\right|},$$

  where, τ signifies the radius of the ball around $\mathcal{E}(a)$.
- **kNN Smoothing.** In this setting, we define a binary similarity function as the k-nearest neighbors decision function:

$$f_\tau\left(\mathcal{E}(a), \mathcal{E}(a')\right) \triangleq \frac{\mathbb{I}\left(a' \in \text{kNN}(a, \tau)\right)}{\tau},$$

  where, τ signifies the number of nearest neighbors to use, and $\text{kNN}(a, \tau)$ represents the set of τ nearest neighbors of $\mathcal{E}(a)$ in $\mathcal{E}_\mathcal{A}$.

We note that our general Policy Convolution framework encompasses the existing OPE estimators designed for large action spaces [23, 28]. As we will later show in Section 4.2, using the convolution functions proposed in Section 3, PC is able to achieve significantly better performance than existing estimators. More specifically, groupIPS [23] can be generalized as PC-IPS and offCEM [28] as PC-DR, both using a two-depth tree (*i.e.*,fl at clustering) in the tree convolution function, with an additional constraint of $\tau_1 = \tau_2 = 1$ (see Appendix B for a formal generalization). As we will further note in our experiments (Section 4.2): (1) using the kernel and kNN convolution functions tend to perform better than others; and (2) convolving the logging and target policies *differently* (*i.e.*, $\tau_1 \neq \tau_2$) adds a lot offl exibility to PC, leading to much better estimation than either convolving the two policies equally, or convolving only one out of the two policies.

Motivating example. To gain a better intuition of PC, we refer to Figure 1 and construct a four-action, single-context toy example described in Table 1. We conduct OPE using the IPS estimator at various levels of the action-tree, with a sample size of $|\mathcal{D}| = 10$ and repeat the experiment 50k times. The results demonstrate that as we progress to higher levels of the tree (increased pooling), variance decreases, but bias increases. At the leaf level, IPS is unbiased but exhibits high variance. At the top-most level, while variance is the lowest, bias is significantly increased. When $\tau = 2$, we observe the best bias-variance trade-off, leading to the lowest MSE.

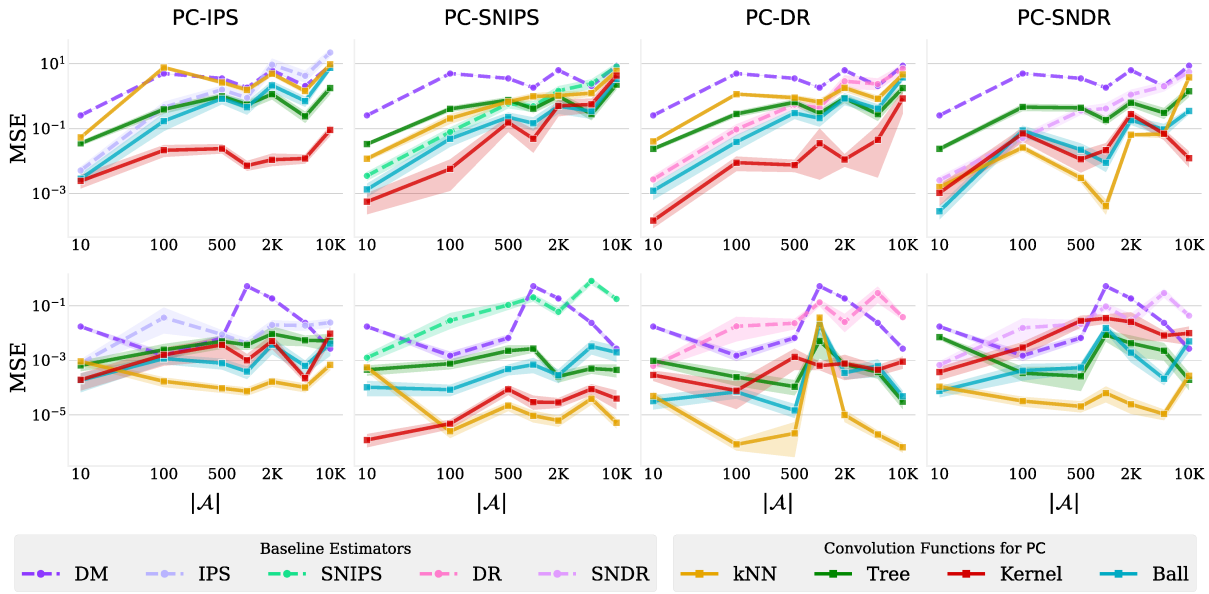

Figure 2: Change in MSE while estimating $V(\pi_{\text{good}})$ with varying number of actions (log-log scale) for the synthetic dataset, using data logged by (top) μ_{uniform}, or (bottom) μ_{good}. Results for $V(\pi_{\text{bad}})$ can be found in supplementary material, Figures 7 and 8.

4 EXPERIMENTS

4.1 Setup

We measure PC's empirical effectiveness on two datasets. Firstly, we simulate a synthetic contextual bandit setup, beginning by sampling contexts $x \sim \mathcal{N}(0, I)$. Subsequently, to realize our assumption that there indeed exists a latent structure amongst individual actions, we randomly assign each action to one of 32 latent *topics*, denoted by $\tilde{a}$. We also assign each latent topic a corresponding mean $\{\mu_i \sim \mathcal{N}(0, I)\}_{i=1}^{32}$ and covariance $\{\sigma_i \sim \mathcal{N}(0, I)\}_{i=1}^{32}$. To realize the assigned structure in the action-space, we sample each action's embedding from its correspondingly assigned topic, *i.e.*, $\mathcal{E}(a) \sim \mathcal{N}(\mu_{\tilde{a}}, \sigma_{\tilde{a}})$. We then model the reward function $\delta(a,x)$ as a noisy and non-linear function of the underlying context- and action-embedding: $\delta(a,x) \triangleq \Phi(x \| \mathcal{E}(a) \| \epsilon)$, where "$\|$" represents concatenation, $\epsilon \sim \mathcal{N}(0, I)$ is white-noise, and Φ is a randomly initialized, two-layer neural network. Such a formulation realizes two crucial assumptions: (a) semantically closer actions are nearby according to $\mathcal{E}$, and (b) $\mathcal{E}$ shares a causal connection with the downstream reward function. Finally, we define the logging policy $\mu(\cdot|x)$ as a temperature-activated softmax distribution on the ground-truth reward distribution $\delta(\cdot, x)$, and the target policy as the $\epsilon-$greedy policy:

$$\begin{aligned} \mu(a|x) &\triangleq \frac{\exp(\beta \cdot \delta(a,x))}{\sum_{a'} \exp(\beta \cdot \delta(a',x))} \\ \pi(a|x) &\triangleq (1-\epsilon) \cdot \mathbb{I}\left(\delta(a,x) = \sup_{a' \in \mathcal{A}} \{\delta(a',x)\}\right) + \frac{\epsilon}{|\mathcal{A}|} \end{aligned} \tag{1}$$

Furthermore, to test the practicality of PC on a real-world, large-scale data, we also synthesize a bandit-variation of the Movielens-100k dataset [7] which consists of numerous (`user,item,rating`) tuples. See Appendix C.1 for a detailed description of the experimental setup for the Movielens dataset.

Due to space constraints, we present further details about the evaluation setup (MSE as the metric, confidence interval computation, *etc.*) as well as experimental design choices about PC-based estimators in Appendix C.2.

Most notably, as per our setup, $V^*(\mu) \propto \beta$ and $V^*(\pi) \propto \epsilon^{-1}$ for both synthetic and Movielens datasets. For clarity, we define μ_{uniform} when $\beta = 0$, and μ_{good} when $\beta = 3$. Similarly, we define π_{bad} when $\epsilon = 0.8$, and π_{good} when $\epsilon = 0.05$. Unless specifically mentioned, we use the default values of the remaining hyper-parameters in the bandit data generation procedure, as listed in Appendix D.

4.2 Results

(Figure 2) How does PC perform with a varying number of actions? Observing the effect of increasing the number of actions ($|\mathcal{A}|$) on different estimators' performance while keeping the size of the available bandit feedback ($|\mathcal{D}|$) constant, wefi nd that the MSE of all IS-based estimators deteriorates, in accordance with the $\Omega(|\mathcal{A}|)$ growth of their variance. We further note that utilizing μ_{good} rather than μ_{uniform} as the logging policy, results in an almost 2-orders of magnitude reduction of MSE across all estimators due to the increased overlap between μ_{good} and π_{good}. Finally, a general trend that holds across various convolution strategies is that the improvement in MSE achieved by PC *w.r.t.* its respective backbone, significantly increases with increasing $|\mathcal{A}|$. Notably, in the extreme scenario of 10k actions, PC-DR outperforms DR by up to 5-orders of magnitude in terms of MSE. While we note that no single backbone estimator or pooling strategy is optimal in every scenario for PC, PC-SNDR and kernel *or* kNN convolution strategies generally exhibit better performance than others.

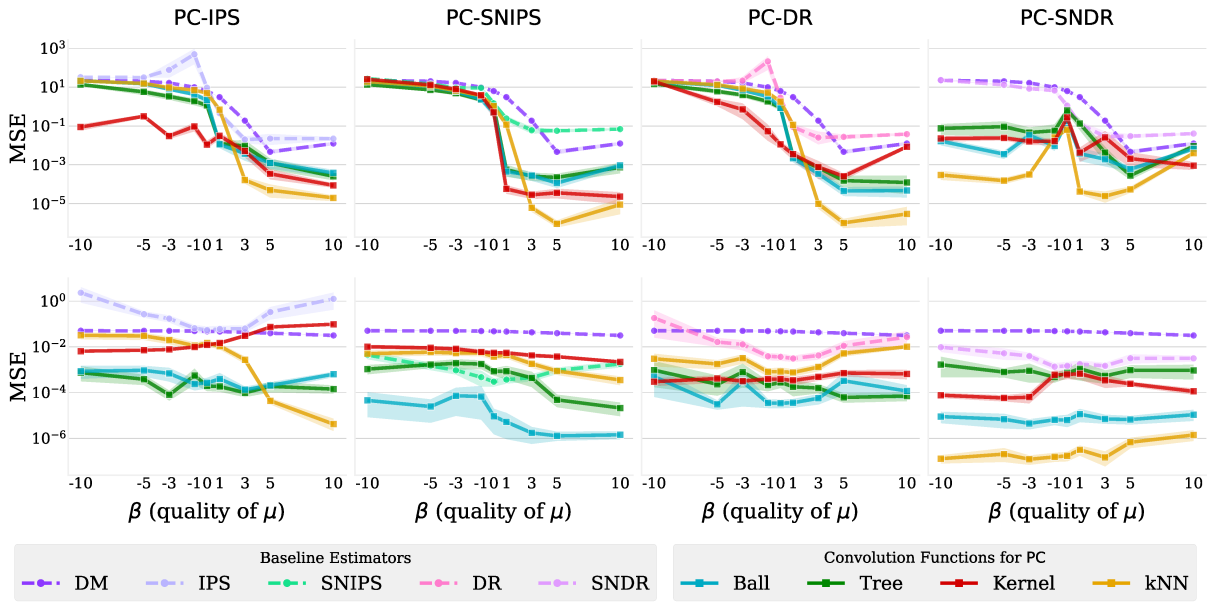

Figure 3: Change in MSE while estimating $V(\pi_{\text{good}})$ with varying policy-mismatch (log scale) for (top) synthetic, and (bottom) movielens dataset. The policy-mismatch is higher when β is lower. Results for estimating $V(\pi_{\text{bad}})$, and the observed bias-variance trade-off can be found in supplementary material, Figures 9 to 12.

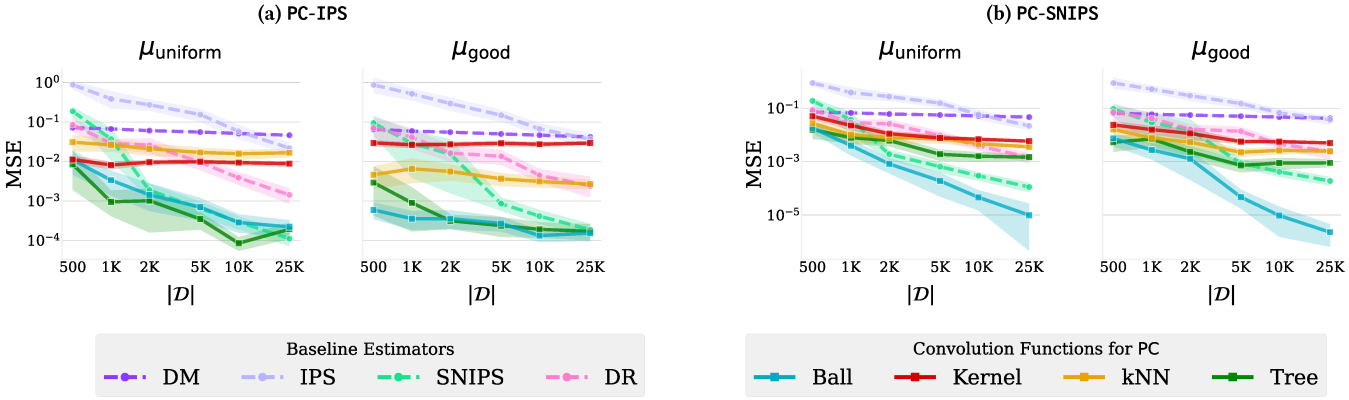

Figure 4: Change in MSE while estimating $V(\pi_{\text{good}})$ with varying amounts of bandit feedback (log-log scale) for the movielens dataset. Results for PC-DR, PC-SNDR, estimating $V(\pi_{\text{bad}})$, and the synthetic dataset can be found in supplementary material, Figures 17 to 20.

(Figure 3) How does PC perform with varying amount of policy-mismatch? We analyze the impact of increasing the policy mismatch between the target and logging policies on off-policy estimation performance, specifically by tuning the β parameter in Equation (1) that controls the quality of the logging policy, keeping the target policyfi xed. We observe that OPE becomes hardest on both ends of the spectrum, *i.e.*, when the logging and target policies have large divergence. The difficulties of OPE in such low-overlap scenarios have been well documented in the literature [1, 13, 26], and we observe that PC—through the use of latent structure amongst actions—is particularly helpful in such conditions. An analysis of the exact bias-variance trade-off provided in supplementary material, Figure 10 reveals that PC is able to effectively (1) reduce the variance when policy-mismatch is low, *i.e.*, $|\beta| \approx 0$, and (2) counteract the bias introduced by IS when Assumption 2.2 is violated, *i.e.*, $|\beta|$ is large. Both of these improvements result in significantly better MSE for PC across the entire policy-mismatch spectrum, as depicted in Figure 3.

(Figure 4) How does PC perform with a varying amount of bandit feedback? Keeping all other factorsfi xed, we investigate the impact of increasing the size of the logged bandit feedback ($|\mathcal{D}|$) on the MSE of various off-policy estimators. Like other baseline estimators, we observe that PC exhibits consistency and is effectively

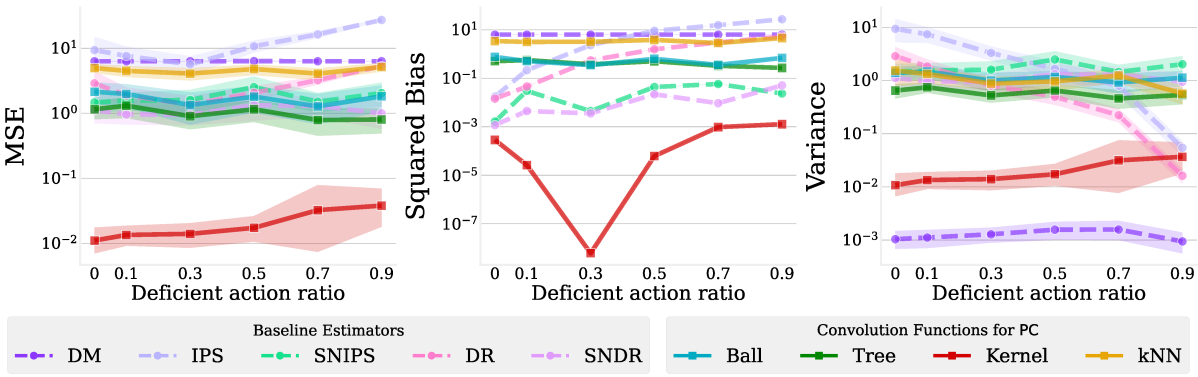

Figure 5: Change in MSE, Squared Bias, and Variance for PC-IPS and other baseline estimators while estimating $V(\pi_{\text{good}})$ with varying support (log-log scale) for the synthetic dataset (with 2000 actions), using data logged by μ_{uniform}. Results for other backbones in PC, and estimating $V(\pi_{\text{bad}})$ can be found in supplementary material, Figures 21 and 22.

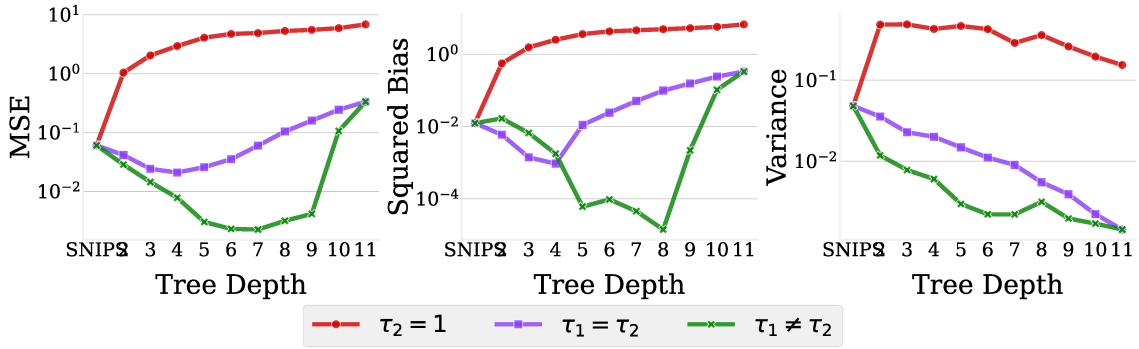

Figure 6: Visualizing the bias-variance trade-offfor PC-SNIPS (Tree pooling) with varying amount of pooling, while estimating $V(\pi_{\text{good}})$ and using μ_{good} for logging on the synthetic dataset (with 2000 actions). When $\tau_1 \neq \tau_2$, we plot results for τ_1 on the plot. The naïve SNIPS estimator is the left-most point, *i.e.*, when there's no pooling. Results for other backbones, pooling methods, and the movielens dataset can be found in supplementary material, Figures 27 and 28.

able to balance the bias-variance trade-off. Specifically, PC is most advantageous in the low-data regime, due to its variance reduction properties. However, as $|\mathcal{D}|$ continues to increase, PC converges to its respective backbone estimator, *i.e.*, when $\tau_1 = \tau_2 = 0$ represents the optimal bias-variance trade-off point. This pattern of a decreasing amount of optimal pooling (τ_1, τ_2) with increasing $|\mathcal{D}|$ is anticipated, as the variance of IS-based estimators naturally decreases with growing $|\mathcal{D}|$, and any reduction in variance at the cost of increased bias negatively impacts the overall MSE. We take further note of this observation for even more kinds of logging policies and backbone estimators in supplementary material, Figures 29 to 31.

(Figure 5) How does PC perform with a varying amount of deficient support? To understand the effect of varying amount of support (or overlap) between the logging and target policies on various estimators, we simulate such a scenario by explicitly forcing the logging policy to only have support (*i.e.*, non-zero probability) over a smaller, random set of actions, and have zero probability for all other actions. Varying this deficient action ratio, wefi rstly observe an expected increase in the MSE and squared bias of baseline estimators like IPS, SNIPS, DR, *etc.* due to the violation of Assumption 2.2, which has been shown to to add irrecoverable bias in importance sampling based estimators [26]. On the other hand, even with an increasing number of deficient actions, the MSE for PC tends to stay relatively constant, with kernel-based convolutions being the best approach. This goes to show that PC is able to accurately leverage action-embeddings as a guide for appropriatelyfilling-in the blind spots while performing OPE.

(Figure 6) How does the amount of convolution affect the bias-variance trade-off? We examine the influence that the amount of convolution in PC has on the bias-variance trade-off for three variations of PC with the tree pooling function: (1) only the target policy is convolved, *i.e.*, $\tau_2 = 1$ which is equivalent to the similarity estimator [6]; (2) both the logging and target policies are convolved equally, *i.e.*, $\tau_1 = \tau_2$ which is equivalent to the offCEM [28] and groupIPS [23] estimators; and (3) both the logging and target policies are convolved and τ_1, τ_2 can be different. From Figure 6, we observe that the amount of convolution results in a bias-variance

trade-off, where larger pooling leads to decreased variance, but increased bias. It is worth mentioning that the initial decrease in bias with convolution is due to the use of μ_{good} for logging, which partially violates Assumption 2.2. This results in a biased SNIPS estimate, which the PC family of estimators are able to effectively recover. Further, we note that solely convolving the target policy (*i.e.*, the similarity estimator) does not necessarily result in a suitable bias-variance trade-off, with the other two convolution strategies being significantly better, and having $\tau_1 \neq \tau_2$ consistently being the best approach.

5 RELATED WORK

Off-policy evaluation. A wide body of literature in operations research, causal inference, and reinforcement learning studies the problem of off-policy evaluation. Prominent off-policy estimators can be grouped into the following three categories: **(1) Model-based:** dubbed as the direct method (DM), whose key idea is to use a parametric reward model to extrapolate the reward for unobserved (context, action) pairs [4]. DM typically has a low variance, at the cost of uncontrollable bias due to model misspecification. **(2) Inverse propensity scoring (IPS)**: IPS uses the propensity ratio between the target and logging policies to account for the distribution mismatch [10]. Though unbiased under mild assumptions, IPS suffers from large variance. Typical remedies for the large variance are propensity clipping [11, 36] or self-normalization [38], which might introduce bias. **(3) Hybrid:** some estimators (e.g, the doubly robust estimator [4]) combine DM and IPS together to leverage the benefits of both worlds [3–5, 24, 34, 42]. However, these estimators still suffer from the large variance problem due to the large propensities especially when the action space is large.

Off-policy evaluation for large action spaces. Two kinds of major problems occur when attempting to perform OPE in large action spaces. Firstly, the variance of any importance sampling method grows linearly *w.r.t.* the size of the action-space [39], and the common support assumption tends to become impractical [26] leading to irrecoverable bias in estimation. Recent work [6, 23, 27, 28] attempts to use some notion of latent structure in the action-space to address both of the aforementioned limitations. The MIPS estimator [27] builds on the randomness in the available action embeddings to improve OPE. However, in a setting where only a bijective mapping between actions and embeddings is available (as in this paper and many real-world scenarios where embedding is in a continuous space), MIPS reduces to vanilla IPS. Further, as we discussed in Section 3, offCEM [28], similarity estimator [6], and groupIPS [23] are all specific instantiations of our PC family of estimators. A concurrent work [41] proposes to use the estimated marginal density ratio over the reward as the importance weight in the estimators.

Off-policy evaluation for continuous action spaces. Another line of work builds off-policy estimators when the action-space is continuous, *e.g.*, the dosage of a treatment. If we discretize the action-space into afi xed number of bins as per some resolution, the action-space becomes too large for typical off-policy estimators to work well [40]. Naive use of importance sampling based estimators would be vacuous in this setting, since the probability of selecting any action can be zero for a policy that samples actions according to some probability density function. To this end, typical approaches extend the discrete rejection sampling idea into a smooth rejection operation using standard kernel functions [14, 17, 45], with the implicit assumption that similar actions (in terms of distance in the continues action space) should lead to similar reward. Our proposed PC also leverages the similarity information between actions through action embeddings, but for problem with discrete and large action spaces.

6 CONCLUSION & FUTURE WORK

In this paper, we proposed the Policy Convolution (PC) family of estimators which leverage latent action structure specified via action embeddings to perform off-policy evaluation in large action spaces. More specifically, PC convolves both the target and logging policies according to an action-action convolution function, which posits a new kind of bias-variance tradeoff controlled by the amount of convolution.

Conducting empirical evaluation over a diverse set of off-policy estimation scenarios, we observe that the estimators from the PC framework enjoy up to 5 orders of magnitude improvement over existing baseline estimators in terms of MSE, especially when (1) the action-space is large, (2) the policy mismatch between logging and target policies is high, or (3) the common support assumption for importance sampling is violated. We believe that ourfindings can expand the potential use of off-policy estimators into new and practical scenarios, and also encourage further exploration into the use of additional structure for efficient OPE.

We also discuss limitations and unexplored directions in this paper that we believe are promising for future work. Firstly, having a deeper formal understanding about the statistical properties of PC might help in designing more robust off-policy estimators. Next, even though we propose four different action convolution functions, having a better understanding of the inductive biases that various convolution functions posit might guide us in designing even better and more principled OPE approaches. Finally, understanding and developing principled techniques for automatically selecting the level of convolution to conduct on the target and logging policies is an interesting research direction [18, 35].

REFERENCES

[1] Jacob Buckman, Carles Gelada, and Marc G Bellemare. The importance of pessimism infi xed-dataset policy optimization. In *International Conference on Learning Representations*, 2021.
[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
[3] Miroslav Dudík, Dumitru Erhan, John Langford, and Lihong Li. Doubly robust policy evaluation and optimization. *Statistical Science*, pages 485–511, 2014.
[4] Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, 2011.
[5] Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. More robust doubly robust off-policy evaluation. In *International Conference on Machine Learning*, pages 1447–1456. PMLR, 2018.
[6] Nicolò Felicioni, Maurizio Ferrari Dacrema, Marcello Restelli, and Paolo Cremonesi. Off-policy evaluation with deficient support using side information. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
[7] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 2015.

[8] Tim Hesterberg. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194, 1995.

[9] Keisuke Hirano and Guido W Imbens. The propensity score with continuous treatments. *Applied Bayesian modeling and causal inference from incomplete-data perspectives*, 226164:73–84, 2004.

[10] Daniel G Horvitz and Donovan J Thompson. A generalization of sampling without replacement from afi nite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

[11] Edward L Ionides. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311, 2008.

[12] Thorsten Joachims, Adith Swaminathan, and Maarten De Rijke. Deep learning with logged bandit feedback. In *International Conference on Learning Representations*, 2018.

[13] Nathan Kallus and Masatoshi Uehara. Efficiently breaking the curse of horizon in off-policy evaluation with double reinforcement learning. *Operations Research*, 2022.

[14] Nathan Kallus and Angela Zhou. Policy evaluation and optimization with continuous treatments. In *International conference on artificial intelligence and statistics*, pages 1243–1251. PMLR, 2018.

[15] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 681–690, 2008.

[16] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[17] Haanvid Lee, Jongmin Lee, Yunseon Choi, Wonseok Jeon, Byung-Jun Lee, Yung-Kyun Noh, and Kee-Eung Kim. Local metric learning for off-policy evaluation in contextual bandits with continuous actions. *Advances in Neural Information Processing Systems*, 35:3913–3925, 2022.

[18] Jonathan N Lee, George Tucker, Ofir Nachum, Bo Dai, and Emma Brunskill. Oracle inequalities for model selection in offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:28194–28207, 2022.

[19] Romain Lopez, Inderjit S Dhillon, and Michael I Jordan. Learning from extreme bandit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[20] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Ji Yang, Minmin Chen, Jiaxi Tang, Lichan Hong, and Ed H Chi. Off-policy learning in two-stage recommender systems. In *Proceedings of The Web Conference 2020*, 2020.

[21] Alberto Maria Metelli, Alessio Russo, and Marcello Restelli. Subgaussian and differentiable importance sampling for off-policy evaluation and learning. *Advances in Neural Information Processing Systems*, 34:8119–8132, 2021.

[22] Anshul Mittal, Noveen Sachdeva, Sheshansh Agrawal, Sumeet Agarwal, Purushottam Kar, and Manik Varma. Eclare: Extreme classification with label graph correlations. In *Proceedings of the Web Conference 2021*, WWW '21. Association for Computing Machinery, 2021.

[23] Jie Peng, Hao Zou, Jiashuo Liu, Shaoming Li, Yibao Jiang, Jian Pei, and Peng Cui. Offline policy evaluation in large action spaces via outcome-oriented action grouping. In *Proceedings of the ACM Web Conference 2023*, 2023.

[24] James Robins, Mariela Sued, Quanhong Lei-Gomez, and Andrea Rotnitzky. Comment: Performance of double-robust estimators when" inverse probability" weights are highly variable. *Statistical Science*, 22(4):544–559, 2007.

[25] Noveen Sachdeva, Mehak Preet Dhaliwal, Carole-Jean Wu, and Julian McAuley. Infinite recommendation networks: A data-centric approach. In *Advances in Neural Information Processing Systems*, 2022.

[26] Noveen Sachdeva, Yi Su, and Thorsten Joachims. Off-policy bandits with deficient support. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, 2020.

[27] Yuta Saito and Thorsten Joachims. Off-policy evaluation for large action spaces via embeddings. In *International Conference on Machine Learning*, pages 19089–19122. PMLR, 2022.

[28] Yuta Saito, Qingyang Ren, and Thorsten Joachims. Off-policy evaluation for large action spaces via conjunct effect modeling. In *international conference on Machine learning*, 2023.

[29] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. Unbiased recommender learning from missing-not-at-random implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020.

[30] Otmane Sakhi, Stephen Bonner, David Rohde, and Flavian Vasile. Blob: A probabilistic model for recommendation that combines organic and bandit signals. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 783–793, 2020.

[31] Rajat Sen, Alexander Rakhlin, Lexing Ying, Rahul Kidambi, Dean Foster, Daniel N Hill, and Inderjit S Dhillon. Top-k extreme contextual bandits with arm hierarchy. In *International Conference on Machine Learning*, pages 9422–9433. PMLR, 2021.

[32] Aleksandrs Slivkins. Contextual bandits with similarity information. In *Proceedings of the 24th annual Conference On Learning Theory*, pages 679–702. JMLR Workshop and Conference Proceedings, 2011.

[33] Aleksandrs Slivkins, Filip Radlinski, and Sreenivas Gollapudi. Learning optimally diverse rankings over large document collections. In *Proc. of the 27th International Conference on Machine Learning (ICML 2010)*, 2010.

[34] Yi Su, Maria Dimakopoulou, Akshay Krishnamurthy, and Miroslav Dudík. Doubly robust off-policy evaluation with shrinkage. In *International Conference on Machine Learning*, pages 9167–9176. PMLR, 2020.

[35] Yi Su, Pavithra Srinath, and Akshay Krishnamurthy. Adaptive estimator selection for off-policy evaluation. In *International Conference on Machine Learning*, pages 9196–9205. PMLR, 2020.

[36] Yi Su, Lequn Wang, Michele Santacatterina, and Thorsten Joachims. Cab: Continuous adaptive blending for policy evaluation and learning. In *International Conference on Machine Learning*, pages 6005–6014. PMLR, 2019.

[37] Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*, pages 814–823. PMLR, 2015.

[38] Adith Swaminathan and Thorsten Joachims. The self-normalized estimator for counterfactual learning. *advances in neural information processing systems*, 28, 2015.

[39] Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miro Dudik, John Langford, Damien Jose, and Imed Zitouni. Off-policy evaluation for slate recommendation. *Advances in Neural Information Processing Systems*, 30, 2017.

[40] Yunhao Tang and Shipra Agrawal. Discretizing continuous action space for on-policy optimization. In *Proceedings of the aaai conference on artificial intelligence*, 2020.

[41] Muhammad Faaiz Taufiq, Arnaud Doucet, Rob Cornish, and Jean-Francois Ton. Marginal density ratio for off-policy evaluation in contextual bandits. In *Neural Information Processing Systems*, 2023.

[42] Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148. PMLR, 2016.

[43] Masatoshi Uehara, Chengchun Shi, and Nathan Kallus. A review of off-policy evaluation in reinforcement learning. *arXiv preprint arXiv:2212.06355*, 2022.

[44] Matt P Wand and M Chris Jones. *Kernel smoothing*. CRC press, 1994.

[45] Lequn Wang, Akshay Krishnamurthy, and Aleksandrs Slivkins. Oracle-efficient pessimism: Offline policy optimization in contextual bandits. In *International Conference on Artificial Intelligence and Statistics*, 2024.

[46] Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudık. Optimal and adaptive off-policy evaluation in contextual bandits. In *International Conference on Machine Learning*, pages 3589–3597. PMLR, 2017.

[47] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983, 2018.

A APPENDIX: FURTHER PC INSTANTIATIONS

For the sake of clarity, we provide a formal definition of PC with using Self-Normalized IPS (SNIPS, Section 2.2.3), Doubly Robust (DR, Section 2.2.4), Self-Normalized DR (SNDR, Section 2.2.5) as backbone estimators:

- **PC-SNIPS.** Defined as follows:

$$\hat{V}_{\texttt{PC-SNIPS}}(\pi) \triangleq \mathop{\mathbb{E}}_{(x,a,r)\sim\mathcal{D}}\left[\frac{(\pi(\cdot|x) * f_{\tau_1})(a)}{\rho \cdot (\mu(\cdot|x) * f_{\tau_2})(a)} \cdot r\right]$$
$$\text{s.t. } \rho \triangleq \mathop{\mathbb{E}}_{(x,a,\cdot)\sim\mathcal{D}}\left[\frac{(\pi(\cdot|x) * f_{\tau_1})(a)}{(\mu(\cdot|x) * f_{\tau_2})(a)}\right].$$

- **PC-DR.** Defined as follows:

$$\hat{V}_{\texttt{PC-DR}}(\pi) \triangleq \mathop{\mathbb{E}}_{(x,a,r)\sim\mathcal{D}}\left[\frac{(\pi(\cdot|x) * f_{\tau_1})(a)}{(\mu(\cdot|x) * f_{\tau_2})(a)} \cdot (r - \hat{\delta}(a,x)) + \Delta(\pi,x)\right]$$
$$\text{s.t. } \Delta(\pi,x) \triangleq \sum_{a'\in\mathcal{A}} \pi(a'|x) \cdot \hat{\delta}(a',x).$$

- **PC-SNDR.** Defined as follows:

$$\hat{V}_{\texttt{PC-SNDR}}(\pi) \triangleq \mathop{\mathbb{E}}_{(x,a,r)\sim\mathcal{D}}\left[\frac{(\pi(\cdot|x) * f_{\tau_1})(a)}{\rho \cdot (\mu(\cdot|x) * f_{\tau_2})(a)} \cdot (r - \hat{\delta}(a,x)) + \Delta(\pi,x)\right]$$
$$\text{s.t. } \rho \triangleq \mathop{\mathbb{E}}_{(x,a,\cdot)\sim\mathcal{D}}\left[\frac{(\pi(\cdot|x) * f_{\tau_1})(a)}{(\mu(\cdot|x) * f_{\tau_2})(a)}\right]$$
$$\text{s.t. } \Delta(\pi,x) \triangleq \sum_{a\in\mathcal{A}} \pi(a|x) \cdot \hat{\delta}(a,x).$$

B APPENDIX: EXISTING OPE ESTIMATORS

We formally show the generalization of existing off-policy estimator, groupIPS [23], which is also designed for large action-spaces to be a specific instantiation of the Policy Convolution framework. Starting with the definition of PC-IPS:

$$\hat{V}_{\texttt{PC-IPS}}(\pi) = \mathop{\mathbb{E}}_{(x,a,r)\sim\mathcal{D}}\left[\frac{\sum_{a'} \pi(a'|x) \cdot f_{\tau_1}(\mathcal{E}(a),\mathcal{E}(a'))}{\sum_{a'} \mu(a'|x) \cdot f_{\tau_2}(\mathcal{E}(a),\mathcal{E}(a'))} \cdot r\right],$$

$\because \tau_1 = \tau_2$ in groupIPS:

$$\hat{V}_{\text{groupIPS}}(\pi) = \mathop{\mathbb{E}}_{(x,a,r)\sim\mathcal{D}}\left[\frac{\sum_{a'} \pi(a'|x) \cdot f_{\tau}(\mathcal{E}(a),\mathcal{E}(a'))}{\sum_{a'} \mu(a'|x) \cdot f_{\tau}(\mathcal{E}(a),\mathcal{E}(a'))} \cdot r\right],$$

Further, $\because f(\cdot,\cdot)$ is a single-depth tree pooling function (equivalent to one-level of clustering) in groupIPS:

$$\hat{V}_{\text{groupIPS}}(\pi) = \mathop{\mathbb{E}}_{(x,a,r)\sim\mathcal{D}}\left[\frac{\sum_{a'} \pi(a'|x) \cdot \mathbb{I}(a' \in \mathbf{a}(a))}{\sum_{a'} \mu(a'|x) \cdot \mathbb{I}(a' \in \mathbf{a}(a))} \cdot r\right],$$

which arrives us to the proposed groupIPS estimator [23], where $\mathbf{a}(a)$ represents the cluster of the action a.

Note that offCEM [28] can be generalized in the exact same way as above, but starting with PC-DR instead of PC-IPS.

C APPENDIX: EXPERIMENTAL SETUP

C.1 Movielens

Taking inspiration from previous recommender system → bandit feedback conversion setups [29], we define a positive reward if the provided rating ≥ 4, or else zero. We then define contexts and action-embeddings as the user- and item-factors attained by performing SVD on the binary user-item rating matrix, respectively. Furthermore, to simulate continuous instead of binary reward, for missing entries, we define the reward as the dot product of the corresponding user- and item-factors, estimated using SVD before. We define the target policy similarly as in Equation (1), and aiming to follow a realistic two-stage recommender system setup [20], we define the logging policy $\mu(\cdot|x)$ as follows: (1) shortlist a set of 100 best actions defined by $\delta(\cdot,x)$, and 400 actions at random; (2) sample a logit from $U(0,1)$ for each positive action, and from $U(0,0.8)$ for the random actions; (3) take a temperature softmax as in Equation (1) only on the sampled logits; and (4) perform $\epsilon-$greedy on the obtained action probabilities to satisfy Assumption 2.2.

C.2 Further Details

For evaluating the performance of various estimators, we compute the Mean Squared Error (MSE) between the true and predicted value of the target policy. We reserve a large test-set just to compute the true value of the target policy. We also estimate the squared bias and variance of our predicted estimates by repeating each experiment for 50 random seeds, and also compute the 95% confidence interval for visualization purposes. Note that the bias, variance, and MSE of any estimator are naturally linked to each other by the following decomposition: $\text{MSE}(\cdot) = \text{Bias}(\cdot)^2 + \text{Var}(\cdot)$.

For estimators in the PC framework, we chose the optimal convolution values (*i.e.*, τ_1 and τ_2) using the MSE obtained on a validation set. Notably, while PC for any given backbone estimator strictly contains the naïve backbone (*i.e.*, when $\tau_1 = \tau_2 = 0$); to de-confound the effect of policy convolution and the backbone estimator, we only report results for PC with a non-zero amount of pooling.

D APPENDIX: DEFAULT HYPERPARAMETERS

Hyper-parameter	Default Value
$\lvert\mathcal{A}\rvert$	2,000
$\lvert\mathcal{D}\rvert$	10,000
| Test data |	100,000
β	0.0
ϵ	0.05
# Seeds	50
dim(context)	32
dim(action-embed)	16
dim(noise)	8