Investigating Action Embeddings for More Efficient Off-Policy Evaluation

Maxime Wabartha*
Kevin H. Wilson
R. David Evans
Hossein Sharifi-Noghabi
Tristan Sylvain
maxime.wabartha@mail.mcgill.ca
RBC Borealis
Canada

Abstract

Off-policy evaluation is known to suffer from high variance in large action spaces. Recent estimators leverage existing structure to reduce the problem dimensionality by using action embeddings. Yet, which properties lead embeddings to be useful for downstream evaluation remains an open question. To answer it, we benchmark several embeddings in a variety of synthetic environments. We observe that even if they exist, the causal action embeddings may not lead to the lowest error in downstream estimation. We then analyze the sensitivity of our findings through several ablations, and highlight that the presence of flat, redundant regions in the reward function, as well the dependency of embeddings on the reward, are key to reducing the variance of embeddings-based estimators. All code and data to reproduce our results will be publicly available.

ACM Reference Format:
Maxime Wabartha, Kevin H. Wilson, R. David Evans, Hossein Sharifi-Noghabi, and Tristan Sylvain. 2025. Investigating Action Embeddings for More Efficient Off-Policy Evaluation. In . ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Algorithmic decisions such as the choice of an item to display, the format of an e-mail to send, or the selection of an advertisement to serve shape the experience of users during online interactions. Personalized choices help ensure quality recommendations, which in turn lead to good user engagement with the proposed content.

Large quantities of logged data are generated from interactions with past policies. Crucially, practitioners can evaluate the performance metrics of new candidate policies on this logged data without having to execute a costly and lengthy A/B test [16]. Common estimators in this setting, known as off-policy evaluation (OPE, [19]), suffer however from exceedingly high variance. Large action spaces are a common cause to this drawback, either because the number of items to be recommended are plentiful, as is often the case in e-commerce recommendation, or because the action space itself becomes combinatorially large [27].

Recent methods solve this problem by leveraging the structure over the actions using action embeddings similarity [13, 21]. When the embeddings compress the action information, they help reduce the variance of the so-called embeddings estimator. Clusters or hierarchies over the actions can serve as a discrete embedding [6, 21, 23]. This kind of structure, ubiquitous in the web, may yet only be a partial observation of the true latent structure. At the same time, an estimator using continuous action embeddings aligned when their rewards are correlated, PC-IPS, has recently been explored [20].

How to devise these embeddings, however, is not clear. Fully or partially observed causal action embeddings directly conditioning the reward or matrix factorization embeddings have both been studied as suitable candidates, but the criterion for this choice remain elusive. Moreover, the exact properties of action embeddings which enable a lower OPE error are uncertain. In this work, we investigate further the links between the properties of the action embeddings and the performance of the PC-IPS estimator.

Contributions. Our contributions are the following:

- We rigorously benchmark embedding estimators whose embeddings have access to different levels of information, varying the amount of action structure and the reward functions classes in the testing environments,
- We shed light on the primacy of the reward information in designing low-variance embeddings estimators.

2 Related works

Dealing with variance. In OPE, the biggest challenge to obtaining a low error is perhaps controlling the variance of the policy value estimator. Several estimators [3, 5, 20, 22, 27] inherit the importance weighting scheme of the reward from the IPS estimator [7, 9]. As a consequence, their variance explodes in the case of large action spaces or in presence of rare actions (according to the logging policy). Control variates methods lower the variance of the OPE in a problem instance-specific manner [2, 10, 30, 31]. Thresholding [3], shrinking [24] or normalizing [26] the importance weights are additional tools to mitigate the variance issue.

*Corresponding author. Work done during an internship.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference'17, Washington, DC, USA
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Structure on the action space. In practice, the existing structure over the actions can be leveraged through *action embeddings* to handle large action spaces. MIPS [21] computes the importance weights *w.r.t.* discrete or continuous stochastic embeddings. MIPS was shown to (1) provably lower variance when the reward is directly conditioned by the causal embedding, and (2) trade-off favorably bias and variance when partially observing the causal embedding. It belongs to a more general family of estimators centered on compressing the information from the reward function [28]. OffCEM [23] and LIPS [13] are respectively a doubly robust and a trainable embedding variant of MIPS. On the contrary, PC-IPS [20] convolves together the reward of actions with similar continuous deterministic embeddings. The continuous formulation expresses partial action similarities, and thus partial contributions to the estimation. Continuous embeddings also open the door to using pre-existing representations of actions, potentially trained without a reward signal, such as LLM embeddings of textual actions. In addition, we remark that recent works have focused on estimating the value of stochastic policies in continuous action spaces, where the action *is* the embedding, using kernels [11, 14, 32]. Finally, the idea of structure has also been developed for online bandits [1, 6].

3 Methods

Notations. We denote the real line $\mathbf{R}$. $\mathcal{P}(\mathcal{S})$ refers to the set of probability distributions over the set $\mathcal{S}$.

3.1 Off-policy evaluation in bandits

Let $\mathcal{X}$ a context space, $\mathcal{A}$ an action space. Let $R : \mathcal{X} \times \mathcal{A} \to \mathcal{P}(\mathbf{R})$ a (stochastic) reward distribution. We denote the expected reward given a context and action $q(x, a) = \mathbf{E}[R(x, a)]$. Without loss of generality, we consider a *causal embedding* function $\mathcal{E}$ such that $q(x, a) = \tilde{q}(x, \mathcal{E}(a))$. A policy π maps $\mathcal{X}$ to $\mathcal{P}(\mathcal{A})$. Contexts are sampled from a distribution $p(x)$, while actions are sampled from policy $\pi : \mathcal{X} \to \mathcal{P}(\mathcal{A})$ as $a \sim \pi(x)$ with probability $\pi(a|x)$, abusing notation. The value of a policy π is $V(\pi) = \mathbf{E}_{X\sim p, A\sim\pi}[q(X, A)]$.

π_0 collects a dataset $D_0 = \{(x_i, a_i, r_i)\}_{i=1}^N$ under the probability distribution $p(x)\pi_0(a|x)p(r|x, a)$. Off-policy evaluation aims at evaluating the value $V(\pi)$ of an arbitrary target policy π from D_0. Inverse propensity scoring (IPS) [9] is (under some assumptions) an unbiased estimator of $V(\pi)$: $\hat{V}_{\mathrm{IPS}}(\pi) = \frac{1}{N}\sum_{i=1}^N \frac{\pi(a_i|x_i)}{\pi_0(a_i|x_i)} r_i$.

Estimators are traditionally compared using the mean-squared error (MSE) criterion *w.r.t.* $V(\pi)$, which decomposes into the sum of squared bias and variance [8]. It is often desirable to design estimators that trade-off some bias for a bigger variance reduction [21].

3.2 Information sharing in large action spaces

To handle the large importance weights of estimators derived from IPS, we may compute importance weights *w.r.t. action embeddings*. Intuitively, embeddings should be "similar" when the actions they represent lead to close rewards for all contexts [20]. The probabilities of similar actions are then aggregated, reducing the adverse effects previously mentioned. This idea is concretized by an abstract estimator (Eq. Sim) defined *w.r.t.* an *observed embedding* ϕ:

$$\hat{V}_{\mathrm{Sim}}(\pi) = \frac{1}{N}\sum_{i=1}^{N} \frac{\sum_{a\in\mathcal{A}} \pi(a|x_i)\mathrm{sim}(\phi(a), \phi(a_i))}{\sum_{a\in\mathcal{A}} \pi_0(a|x_i)\mathrm{sim}(\phi(a), \phi(a_i))} r_i, \qquad \text{(Sim)}$$

which implements both MIPS and the kernel version of PC-IPS (with bandwidth h) for sim respectively $1_{\phi(a)=\phi(b)}$ and $k_h(\phi(a), \phi(b))$.

What properties for ϕ? Action structure might exist without the algorithm designer being able to observe $\mathcal{E}$ at all, on the contrary to the experiments of MIPS and PC-IPS. We then need to choose design guidelines for ϕ, a question that was not investigated by the previous works [20, 21]. We restrict our study to continuous, deterministic embeddings. First, we focus on whether $\mathcal{E}$ is a reference that ϕ should aim to approximate. Else, the remaining signal to leverage to design ϕ is the reward. Then, what should be the link between ϕ and r? Are embeddings reflective of the general similarities between actions, beyond the reward itself, a good choice?

Benchmark. To answer these questions, we benchmark PC-IPS on independent (unstructured) and groups (structured) causal embeddings to illustrate different levels of action structure. We also explore two reward function classes, namely a dot-product and non-linear neural reward functions. We vary the observed embeddings ϕ used by PC-IPS to estimate the policy value (precise methodology described in App. A.) designed with different levels of information, ranging from structure-agnostic embeddings and reward informed matrix factorization (MF) embeddings to the privileged $\mathcal{E}$. To evaluate the characteristics of MF embeddings, we learn them using a dedicated external fully-observed, noiseless dataset of context, action and rewards. This enables us to focus on the embedding properties that lead to good estimation with PC-IPS, irrespective of the issues inherited from small sample sizes or noise.

4 Experiments

We focus our attention on answering the following research question: **what aspects of the embeddings and of the reward geometry are responsible for reducing the MSE of PC-IPS?**

4.1 Matrix factorization embeddings can improve PC-IPS over causal embeddings

The main results of the benchmark are illustrated in Fig. 1, and we push to the appendix additional ablations on the sample size (App. B.2), the number of actions (App. B.4), the embedding dimension (App. B.3) and the reward slope (App. B.5).

When action structure is present (bottom of Fig. 1), MIPS (who has access to the true structure) unsurprisingly performs best. PC-IPS with causal and MF embeddings then perform similarly and outperform the naive IPS and PC-IPS with independent embeddings.

More interestingly, in the absence of causal embeddings structure (top two subplots), PC-IPS using MF embeddings outperforms both IPS and PC-IPS using the causal embeddings, even though the latter are causally predictive of the reward and the considered Lipschitz rewards ensure that actions with close embeddings have close rewards [20]. The MF embeddings thus extract action similarity signal from the reward function itself, even with uninformative causal embeddings. This conclusion (also observed by Cief et al. [4]) pushes us to investigate further the reasons for this performance improvement through two complementary questions: what characteristics (1) of the environment and (2) of the learned embeddings are conducive of low downstream estimation MSE?

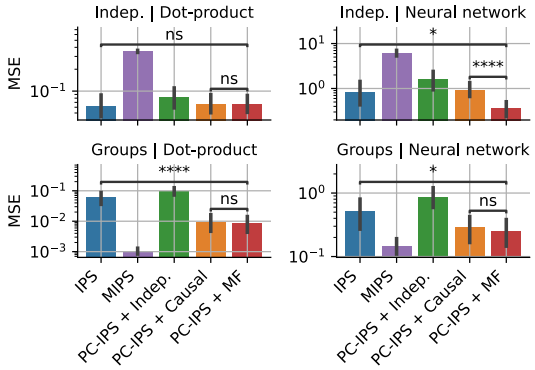

Figure 1: "Classic" benchmark (10k samples, 1024 actions), varying causal embeddings (rows) and reward functions (columns). Lower is better. Statistical significance highlighted with $*$ (see App. A.1.)

4.2 Benefits from flat regions of the reward

We show that flat regions of the reward function are key to lower PC-IPS variance, using a simple setup where the reward is a scaled dot-product, $q(x, a) = \alpha\langle x, \mathcal{E}(a)\rangle$. Fig. 2 shows that PC-IPS outperforms IPS when the reward function is approximately flat (for $\alpha \approx 0$), but approaches IPS in the groups case as α increases. Performance declines at higher reward scaling, suggesting MF embeddings share information across actions with significantly different rewards, itself leading to an increase in bias as shown in App. B.5 (Fig. 13 and Fig. 14).

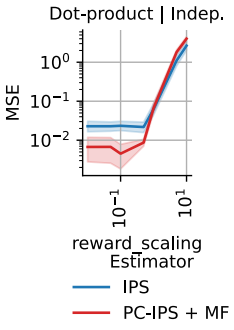

Figure 2: Ablating the reward slope by scaling.

4.3 Matrix factorization generates clusters of action embeddings

While MF can recover the causal embeddings for dot-product rewards , we observe that the algorithm naturally assigns similar embeddings to actions with similar rewards across contexts. Indeed, for any two actions a and a', the projection relationship must satisfy $r(x, a) \approx \langle\psi(x), \phi(a)\rangle \approx \langle\psi(x), \phi(a')\rangle \approx r(x, a')$. We distinguish two mechanisms driving the effectiveness of MF embeddings. The *constructive effect* leads MF to extract aligned embeddings when the causal embeddings exhibit inherent structure (Fig. 15). The *deductive effect* aligns MF embeddings for actions embedded in low reward curvature regions, regardless of the true structure.

To isolate these effects, we systematically increase the intra-cluster variance of group action embeddings datasets (Fig. 3), thus weakening the constructive effect. Yet, MF embeddings maintain strong performance even at high intra-cluster variance, confirming the importance of the deductive effect.

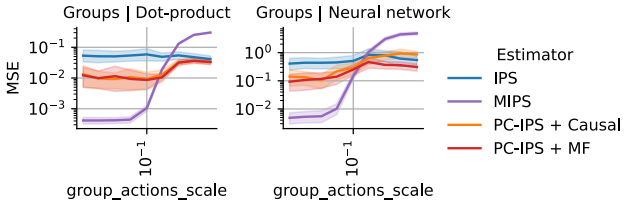

Figure 3: Ablating the structure of the embeddings. A higher group actions scale leads to high intra-group variance.

4.4 Risks of non reward-centric embeddings

Constructive and deductive effects can be unified when causal embeddings exist, as actions within low-reward curvature clusters naturally yield similar rewards. This distinction has a more practical effect when practitioners use arbitrary pre-existing action embeddings encoding action similarity, such as LLM embeddings. Actions with semantic differences (therefore with unaligned LLM embeddings) might impact identically the reward for a specific task.

We design an experiment where actions are described by two clusters: "reward" clusters (determining the reward) and twice as many "observed" clusters representing non reward-centric (NRC) embeddings which fail to capture the reward-relevant compression of the action space. Fig. 4 shows that MF embeddings leverage the invariance of the reward to obtain a lower MSE, while NRC embeddings cannot do so and perform worse with increasing number of clusters. We analyze further this result in App. B.1.

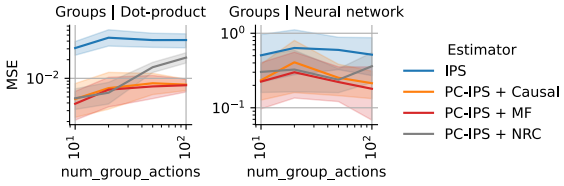

Figure 4: NRC embeddings fail to reduce variance. Using a fixed number of actions leads to more a complex task with increasing numbers of observed clusters `num_group_actions`.

5 Discussion and conclusions

In this document, we delved into the characteristics of embeddings that help PC-IPS during OPE. Through several experiments and ablations, we observed the importance of choosing embeddings that are informed by the reward function, such that the curvature of the reward function *w.r.t.* the chosen embeddings is low. In the future, a self-supervised learning loss [12] could help better match reward similarity and embeddings similarity [14]. In addition, a theoretical analysis quantifying exactly how much two actions leading to similar – but not identical – rewards should interact would be an interesting follow-up.

References

[1] Aouali, I., Kveton, B., and Katariya, S. Mixed-effect thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pp. 2087–2115. PMLR, 2023.

[2] Bibaut, A. F., Malenica, I., Vlassis, N., and van der Laan, M. J. More Efficient Off-Policy Evaluation through Regularized Targeted Learning, December 2019.

[3] Bottou, L., Peters, J., Quiñonero-Candela, J., Charles, D. X., Chickering, D. M., Portugaly, E., Ray, D., Simard, P., and Snelson, E. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research*, 14(1):3207–3260, 2013.

[4] Cief, M., Golebiowski, J., Schmidt, P., Abedjan, Z., and Bekasov, A. Learning action embeddings for off-policy evaluation. In *European Conference on Information Retrieval*, pp. 108–122. Springer, 2024.

[5] Dudík, M., Langford, J., and Li, L. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*, 2011.

[6] Hong, J., Kveton, B., Katariya, S., Zaheer, M., and Ghavamzadeh, M. Deep hierarchy in bandits. In *International Conference on Machine Learning*, pp. 8833–8851. PMLR, 2022.

[7] Horvitz, D. G. and Thompson, D. J. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47 (260):663–685, 1952.

[8] James, G., Witten, D., Hastie, T., Tibshirani, R., et al. *An introduction to statistical learning*, volume 112. Springer, 2013.

[9] Kahn, H. and Harris, T. E. Estimation of particle transmission by random sampling. *National Bureau of Standards applied mathematics series*, 12:27–30, 1951.

[10] Kallus, N. and Uehara, M. Intrinsically Efficient, Stable, and Bounded Off-Policy Evaluation for Reinforcement Learning, June 2019.

[11] Kallus, N. and Zhou, A. Policy evaluation and optimization with continuous treatments. In *International conference on artificial intelligence and statistics*, pp. 1243–1251. PMLR, 2018.

[12] Keramati, M., Meng, L., and Evans, R. D. Conr: Contrastive regularizer for deep imbalanced regression. *arXiv preprint arXiv:2309.06651*, 2023.

[13] Kiyohara, H., Nomura, M., and Saito, Y. Off-policy evaluation of slate bandit policies via optimizing abstraction. In *Proceedings of the ACM on Web Conference 2024*, pp. 3150–3161, 2024.

[14] Lee, H., Lee, J., Choi, Y., Jeon, W., Lee, B.-J., Noh, Y.-K., and Kim, K.-E. Local metric learning for off-policy evaluation in contextual bandits with continuous actions. *Advances in Neural Information Processing Systems*, 35:3913–3925, 2022.

[15] Lepskii, O. On a problem of adaptive estimation in gaussian white noise. *Theory of Probability & Its Applications*, 35(3):454–466, 1991.

[16] Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670, 2010.

[17] Mairal, J., Bach, F., Ponce, J., and Sapiro, G. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pp. 689–696, 2009.

[18] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] Precup, D., Sutton, R. S., and Singh, S. Eligibility traces for off-policy policy evaluation. In *ICML*, volume 2000, pp. 759–766. Citeseer, 2000.

[20] Sachdeva, N., Wang, L., Liang, D., Kallus, N., and McAuley, J. Off-policy evaluation for large action spaces via policy convolution. In *Proceedings of the ACM on Web Conference 2024*, pp. 3576–3585, 2024.

[21] Saito, Y. and Joachims, T. Off-policy evaluation for large action spaces via embeddings. In *International Conference on Machine Learning*, pp. 19089–19122. PMLR, 2022.

[22] Saito, Y., Aihara, S., Matsutani, M., and Narita, Y. Open bandit dataset and pipeline: Towards realistic and reproducible off-policy evaluation. *arXiv preprint arXiv:2008.07146*, 2020.

[23] Saito, Y., Ren, Q., and Joachims, T. Off-policy evaluation for large action spaces via conjunct effect modeling. In *international conference on Machine learning*, pp. 29734–29759. PMLR, 2023.

[24] Su, Y., Dimakopoulou, M., Krishnamurthy, A., and Dudík, M. Doubly robust off-policy evaluation with shrinkage. In *International Conference on Machine Learning*, pp. 9167–9176. PMLR, 2020.

[25] Su, Y., Srinath, P., and Krishnamurthy, A. Adaptive estimator selection for off-policy evaluation. In *International Conference on Machine Learning*, pp. 9196–9205. PMLR, 2020.

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

[27] Swaminathan, A., Krishnamurthy, A., Agarwal, A., Dudik, M., Langford, J., Jose, D., and Zitouni, I. Off-policy evaluation for slate recommendation. *Advances in Neural Information Processing Systems*, 30, 2017.

[28] Taufiq, M. F., Doucet, A., Cornish, R., and Ton, J.-F. Marginal density ratio for off-policy evaluation in contextual bandits. *Advances in Neural Information Processing Systems*, 36:52648–52691, 2023.

[29] Tucker, G. and Lee, J. Improved estimator selection for off-policy evaluation. In *Workshop on Reinforcement Learning Theory at the 38th International Conference on Machine Learning*, 2021.

[30] Vlassis, N., Bibaut, A., Dimakopoulou, M., and Jebara, T. On the Design of Estimators for Bandit Off-Policy Evaluation. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 6468–6476. PMLR, May 2019.

[31] Vlassis, N., Chandrashekar, A., Amat, F., and Kallus, N. Control variates for slate off-policy evaluation. *Advances in Neural Information Processing Systems*, 34: 3667–3679, 2021.

[32] Zenati, H., Bietti, A., Martin, M., Diemert, E., Gaillard, P., and Mairal, J. Counterfactual learning of stochastic policies with continuous actions. *Transactions on Machine Learning Research*, 2025.

A Evaluation procedure

We extend the evaluation methodology from [21], which we explicate in Alg. 1. In our experiments, the environment is defined by the class of the reward function and the structure on the causal embeddings that are used to generate the reward. We consider unstructured causal embeddings, sampled at random, as well as clustered (or groups) causal embeddings, sampled around group representative vectors. We examine two reward function classes: a dot-product between context x and causal action embeddings $\mathcal{E}(a)$, and a randomly initialized neural network operating on concatenated $[x, \mathcal{E}(a)]$, to evaluate PC-IPS behavior across different reward structures. To ensure the generality of our findings, the parameters of the causal embeddings and reward functions are sampled independently at each seed. For each experiment, we also evaluate the estimators on the two static reward functions defined by [14], which can be accessed in the appendix.

To be able to evaluate the characteristics of embeddings leading to low error with PC-IPS, we learn the embeddings using an external fully-observed, noiseless dataset of context, action and rewards. This enables us to focus on the embeddings properties that lead to good estimation with PC-IPS, irrespective of the issues inherited from small sample sizes or noise. Note that the external dataset corresponds to prior knowledge that is only accessed to learn the embeddings. The estimation of the policy is then done using the sampled bandit feedback accessible to all estimators, as detailed in Alg. 1.

Algorithm 1 Evaluation procedure

Require: External dataset $D_{\text{ext}} = \{(x, a, r)\}$
Define: Distribution $p(x)$
Define: Causal action embeddings $\mathcal{E}(a)$
Define: Parametrized reward function class f
Learn: Action embeddings $\phi(a) = \phi(a; D_{\text{ext}})$
for $s = 1$ **to** num seeds **do**
  Sample: Reward function parameters $\theta^{(s)}$, s.t. $q^{(s)}(x, a) = f(x, \mathcal{E}(a); \theta^{(s)})$
  Define: Logging policy $\pi_0^{(s)}(x, a) = \text{softmax}_\beta(q^{(s)}(x, a))$
  Define: Target policy $\pi^{(s)}(x, a) = \epsilon\text{-greedy}(q^{(s)}(x, a))$
  Compute: Target policy value $V(\pi^{(s)}; D_{\text{eval}}))$
  Sample: Bandit feedback $D_s = \{(x_i, a_i, r_i)\}_{i=1..n}$
  Define: Cross-fitted regression model $\hat{q}(x, a; D_s)$
  for each estimator est **do**
    Compute: $\hat{V}_{\text{est}}(\pi^{(s)}; D_s)$
    Compute: $||V(\pi^{(s)}) - \hat{V}_{\text{est}}(\pi^{(s)}; D_s)||_2^2$
  end for
end for

Table 1: Default environment hyperparameters.

Hyperparameter	Value
Reward variance	0.5
β (uniformity of logging policy)	0.1
ϵ (prob. of taking highest value action)	0.05
Context dimension d_x	10
Number of actions	1024
Ratio of deficient actions	0.0
Number of clusters (action groups)	10
Dispersion of action groups σ_g^2	0.1
Reward slope α	1

A.1 Additional methodological details

Contexts are sampled uniformly over a d_x-dimensional sphere (they are therefore of norm 1). When the causal embeddings are group embeddings, actions are first assigned at random to a group associated to a group representative vector. Then, the action causal embedding is sampled from a normal distribution centered on the group representative vector and of variance σ_g^2. We use in practice the `scikit-learn` [18] implementation of the "dictionary learning" algorithm [17] to learn the matrix factorization embeddings extracted from the reward function.

Table 2: Default embedder hyperparameters.

Hyperparameter	Value
Embedding dimension	10
External dataset D_{ext} number of contexts	1000

Table 3: Default estimation hyperparameters.

Hyperparameter	Value
Sample size	10000
Number of seeds	100
Ground truth estimation sample size	20000

Hyperparameter selection. We select the bandwidth of the PC-IPS estimator using the SLOPE++ [29] algorithm, an extension of SLOPE [25]. SLOPE++ is a non-parametric method inspired from the Lepskii principle [15] and adapted to the problem of hyperparameter selection for OPE.

Statistical methodology. All means are estimated with 100 seeds. We compute the confidence intervals using the bootstrap and default hyperparameters from `seaborn`. In bar plots (Fig. 1 and Fig. 5), we compare methods using a paired t-test following the `statannotations` library. The lowest significance level (*) is set to 0.05, while the following (**, *** and ****) are respectively set to $1\,\text{e}-2$, $1\,\text{e}-3$ and $1\quad\text{e}-4$. A p-value superior to 0.05 is indicated as non significant (ns).

A.2 Comparison between the classical benchmark and the "LML" benchmark

In the following sections, we also record the performance of each estimator on the "LML" benchmark, which refers to two reward functions that were defined by [14]. We now compare the LML benchmark with the classical benchmark that we studied in the main part of this document:

- A "classic" benchmark, where the reward function classes include a dot product reward and a randomly initialized MLP. Contexts are sampled uniformly over a `context_dim` dimensional sphere (they are therefore of norm 1).
- A "LML" benchmark, where we re-use the reward functions defined in [14]. The causal embeddings $\phi(a)$ are 2-dimensional continuous representations of the actions (adapted from the continuous action setting) defined in the 2-dimensional unit cube. Contexts are sampled uniformly in the 2-dimensional unit cube, and have therefore varying norm.

More precisely, the reward functions from the LML benchmark are defined as such:

- Absolute: $q(x, a) = -|x_0 - a_0|$
- Quadratic: $-(x-a)^T \begin{pmatrix} 11 & 9 \\ 9 & 11 \end{pmatrix} (x-a)$

B All results

B.1 Default benchmark (10k samples, 1024 actions)

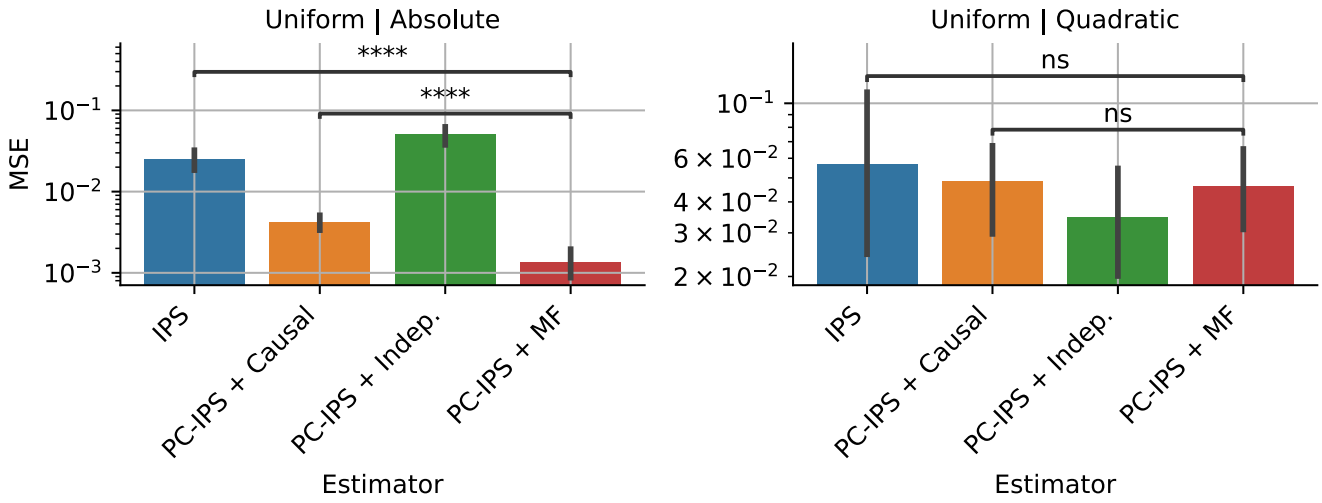

Figure 5: LML benchmark with default configuration (10k samples, 1024 actions).

Analysis. In the left subplot of Fig. 5, we notice that the matrix factorization embeddings significantly outperform both IPS and PC-IPS + causal. We can explain the latter performance improvement by the ability of MF to ignore the irrelevant second coordinate of the context, therefore compressing the information contained in the reward function. We verify this behavior by plotting a T-SNE transformation of the MF embeddings in Fig. 6. In the left subplot, we color each MF embedding with the value of the coordinate of the causal embedding, that is, the only one that has an impact on the reward. We observe that the MF embeddings are organized such that nearby actions in the causal embedding space (and therefore reward) are also mapped nearby in learned embedding space. Conversely, we observe in the right subplot that there is no such organization when coloring according to the second coordinate of the causal embedding, which is irrelevant to the reward. This compression phenomenon additionally illustrates the discussion of Sec. 4.4 about non reward-centric embeddings.

The right subplot of Fig. 5 illustrates a failure of the MF embeddings to improve upon the causal embeddings or even IPS. Including a context-dependent notion of similarity, as proposed by [14], might enable PC-IPS to successfully use the action structure.

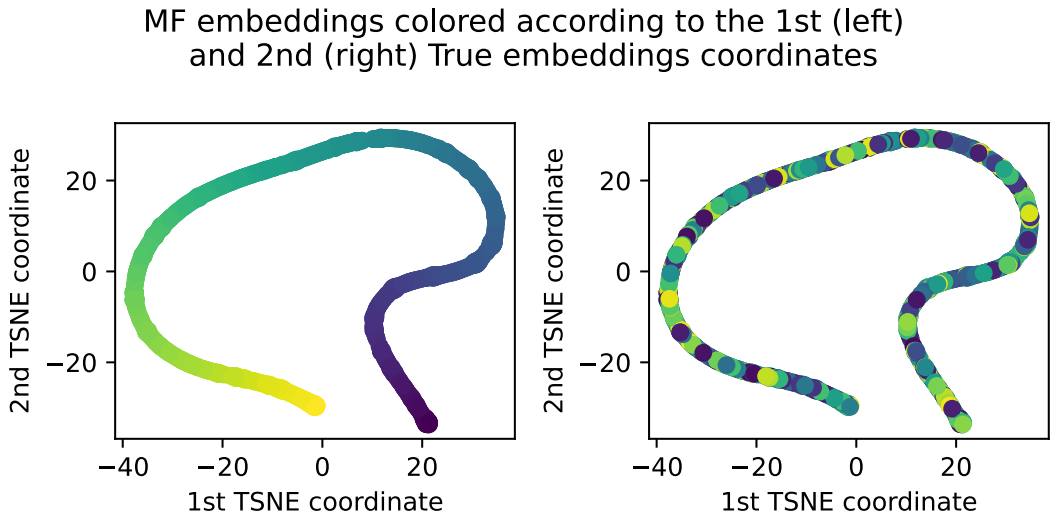

Figure 6: MF embeddings leverage the invariances of the reward function.

B.2 Sample size ablation

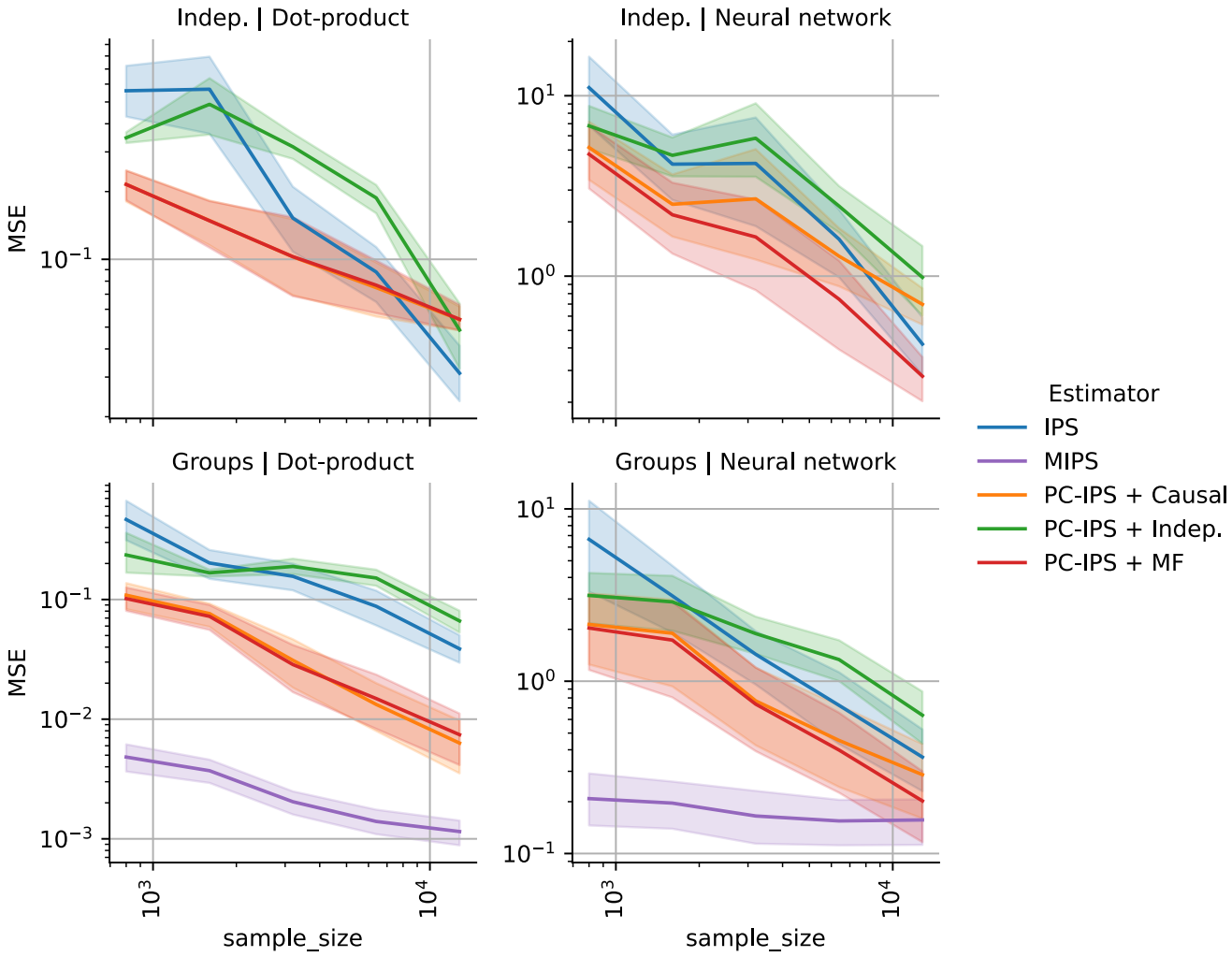

Figure 7: Sample size ablation over the classic benchmark.

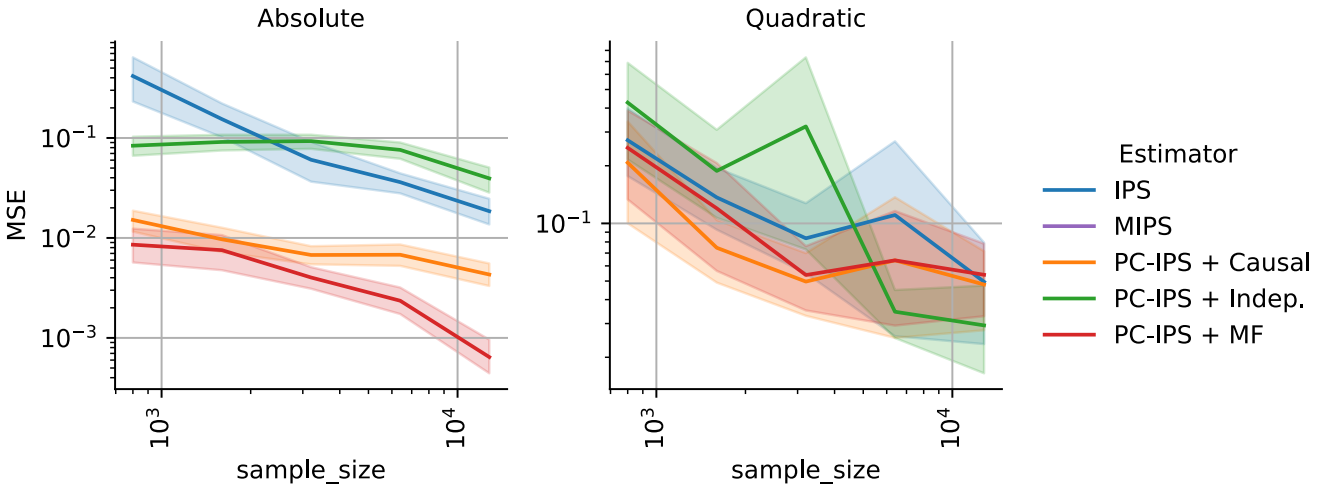

Figure 8: Sample size ablation over the LML benchmark.

B.3 Embedding dim ablation

Approximation quality. Action embeddings may be useful despite significant matrix factorization reconstruction error, provided they respect reward similarity between actions [4]. In particular, embeddings capturing invariance should be extremely useful irrespective of model error. Finally, matrix factorization algorithms require to set an embedding dimension parameter, which influences the OPE performance of the estimator by trading-off some bias when the embedding dimension is too small (underfitting) and variance in the opposite case (overfitting). We push the corresponding ablations in App. B.3 (Fig. 9 and Fig. 10).

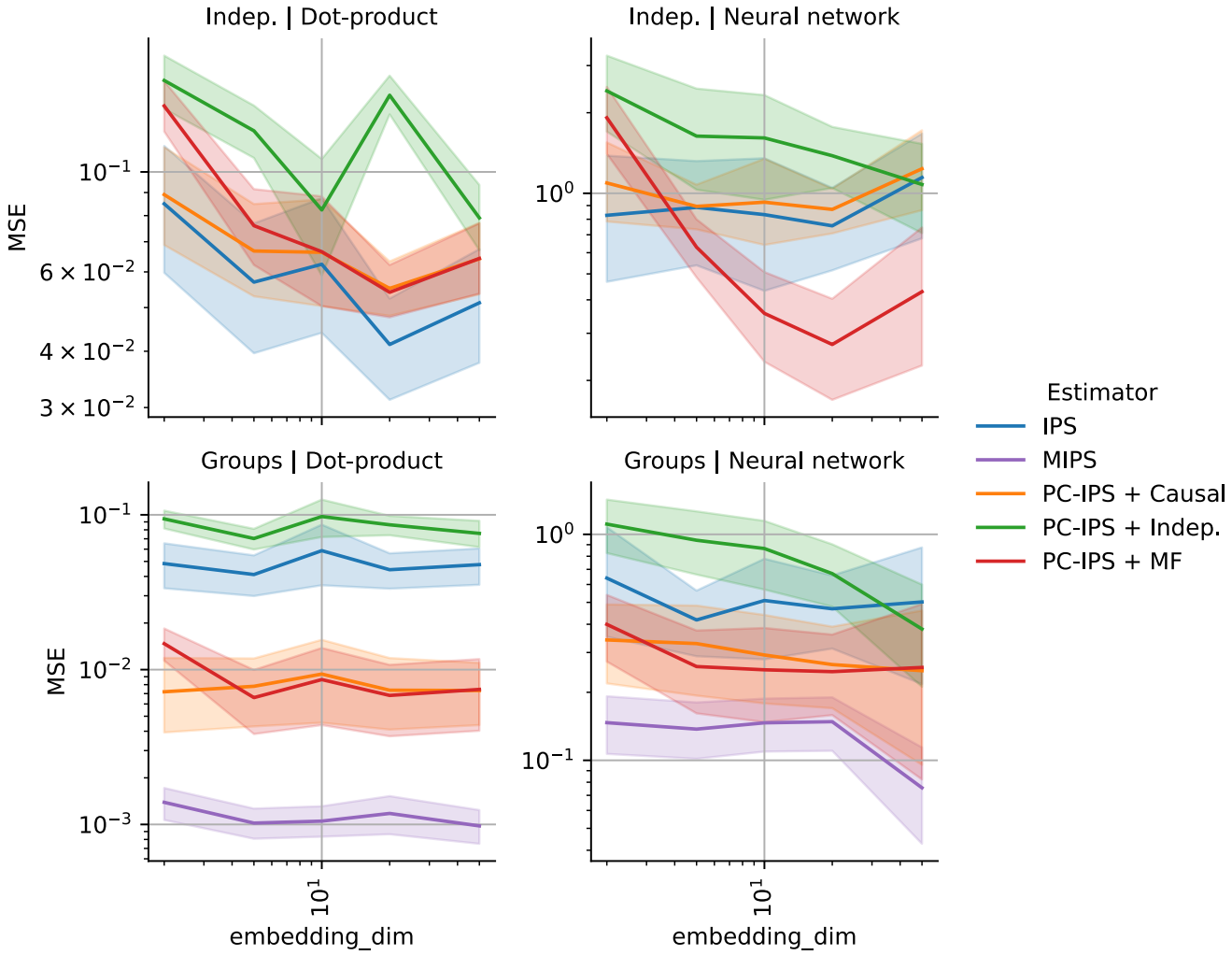

Figure 9: Embedding dimension ablation (used by PC-IPS estimators) over the classic benchmark.

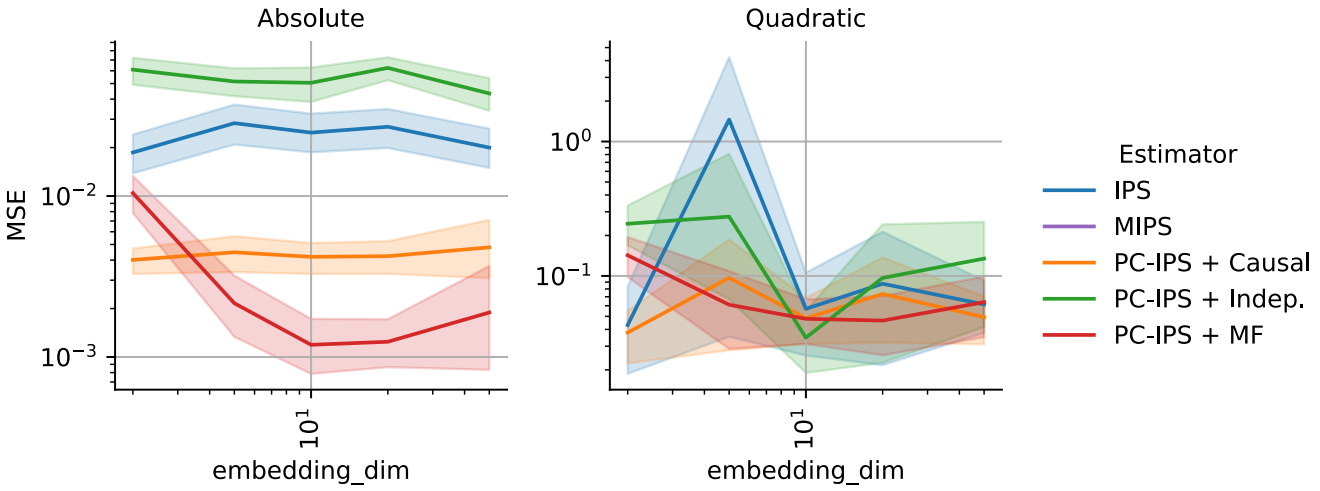

Figure 10: Embedding dimension ablation (used by PC-IPS estimators) over the LML benchmark.

B.4 Action space size ablation

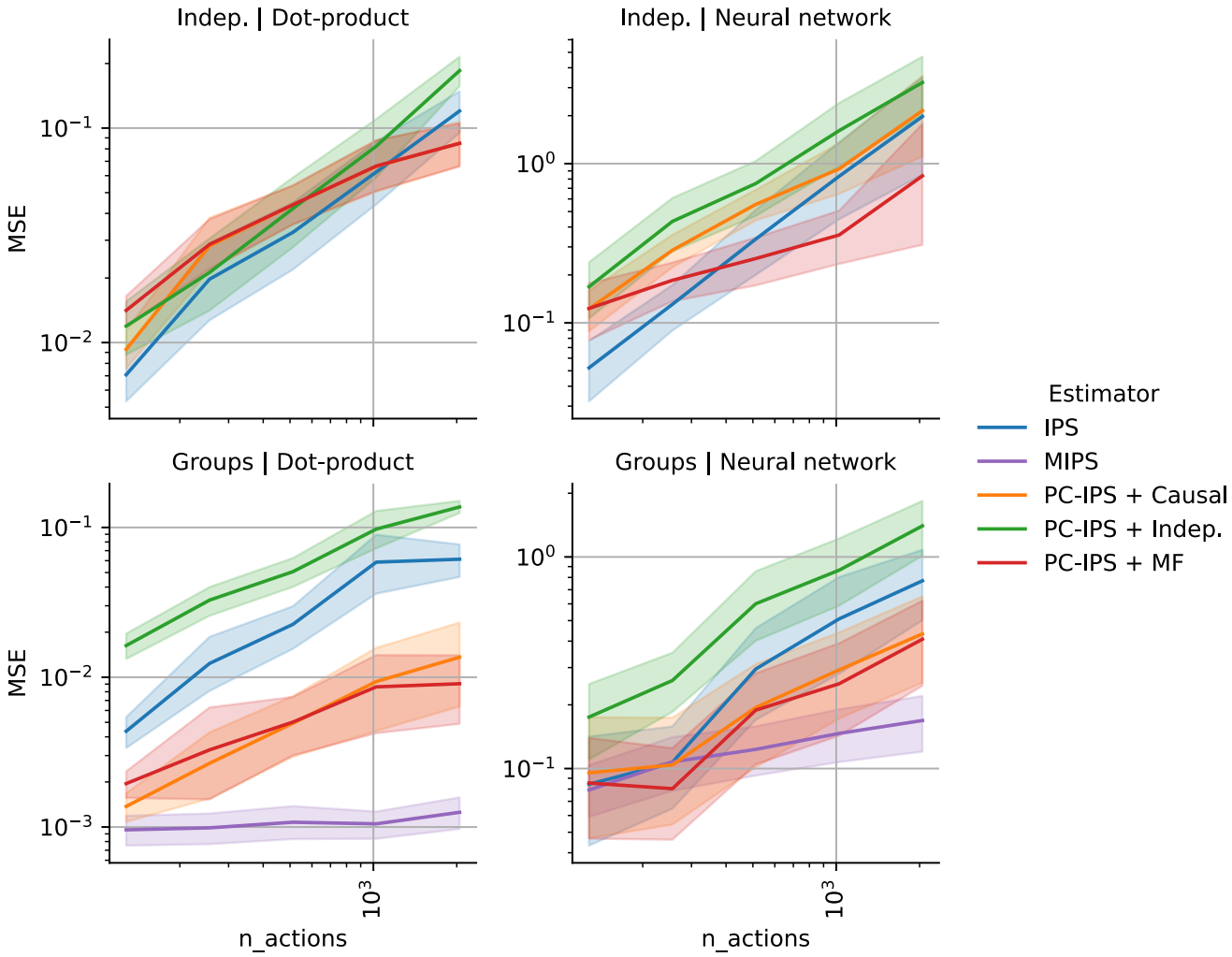

Figure 11: Ablation of the number of actions over the classic benchmark.

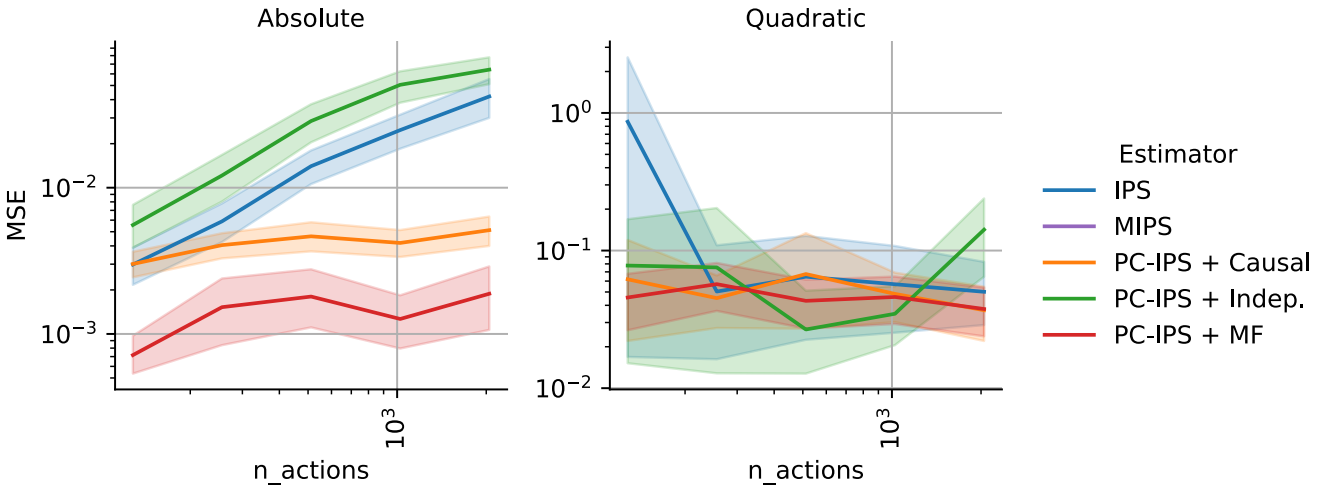

Figure 12: Ablation of the number of actions over the LML benchmark.

B.5 Reward slope ablation

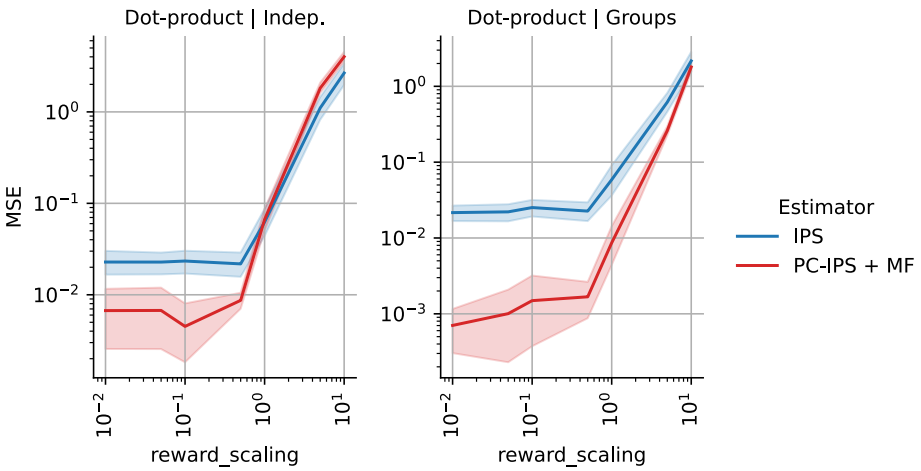

Figure 13: Ablation of the slope of the reward over the classic benchmark. Scale is applied uniformly for all context-action pairs.

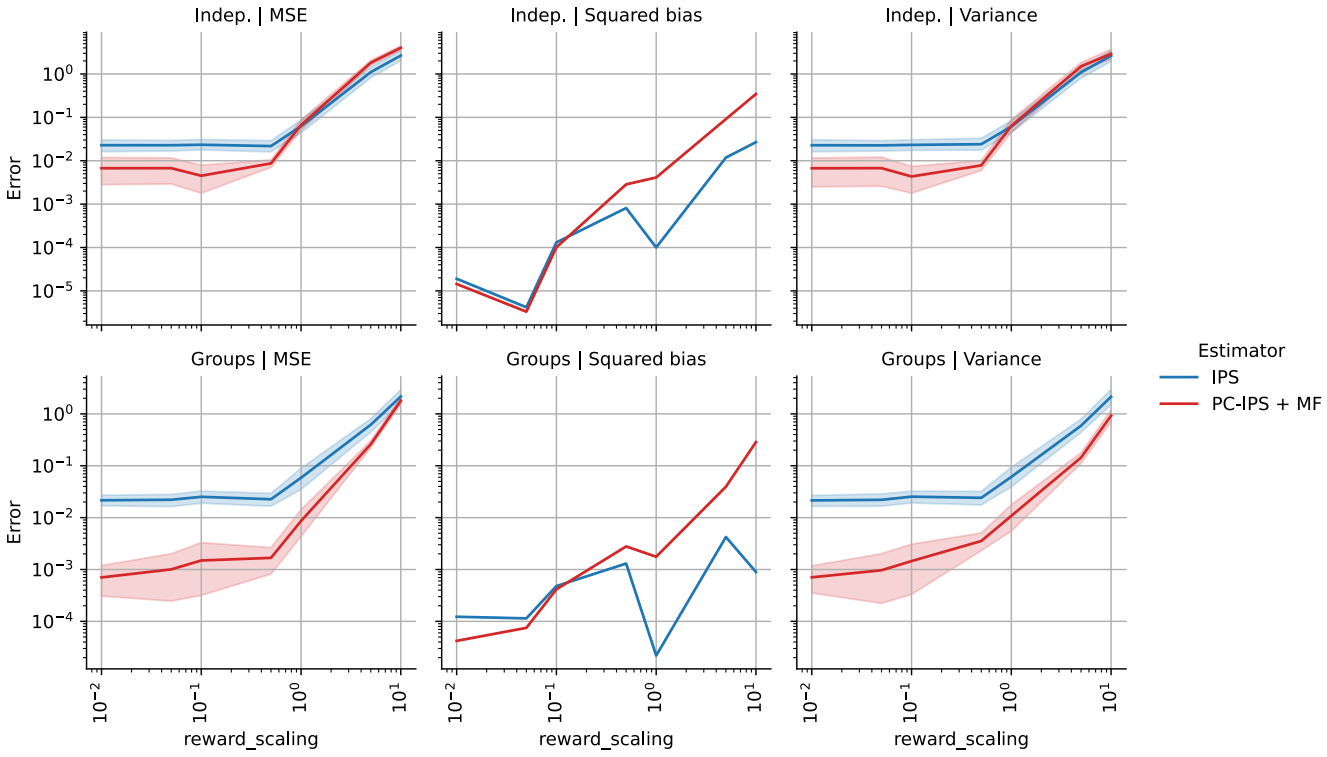

Figure 14: Ablation of the slope of the reward over the classic benchmark. Scale is applied uniformly for all context-action pairs. Bias-variance decomposition.

C Visualizing dictionary learning embeddings

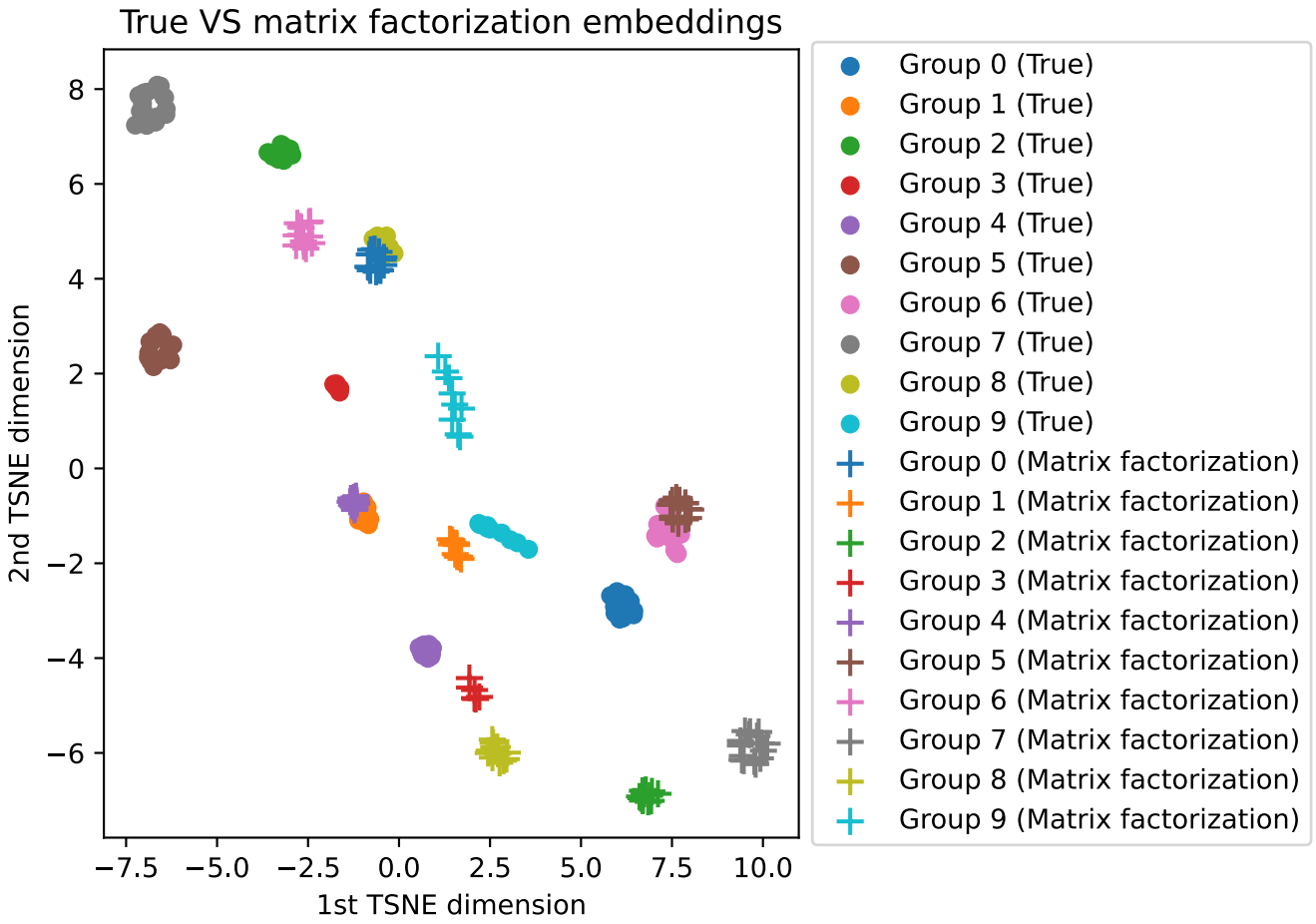

Figure 15: Comparing the causal embeddings (named "True", represented as circles) with the dictionary learned ones (crosses). Matrix factorization learns to cluster together actions leading to the same reward, recovering the true clusters. This explains its great performance used in conjunction with the PC-IPS algorithm.