Off-Policy Learning for Diversity-aware Candidate Retrieval in Two-stage Decisions

Haruka Kiyohara [1] **Rayhan Khanna** [1] **Thorsten Joachims** [1]

Abstract

Two-stage decision systems, which first retrieve a candidate set of items (e.g., fashion items or documents) and then generate the output (e.g., ranked featured items or articles) from the candidates are widely adopted in real-life applications, including search, recommendations, and retrieval-augmented generation (RAG). Diversity in the candidate set is considered a crucial aspect in news recommendation or opinion summarization. However, conventional approaches to candidate retrieval fail to incorporate diversity without post-processing, as they model a single representation of user preference and ignore the (multi-modal) distribution of user preferences on diverse items. To circumvent this issue, we propose a novel *Off-Policy Learning* (OPL) framework that can (1) model the multi-modal distribution of user preference and (2) optimize the preference distribution and candidate set to maximize the user engagement signal, using logged bandit feedback. Moreover, we present a *Kernel Importance Sampling* (Kernel IS)-based policy gradient estimator to mitigate the issues of high variance, deficient support, and severe rejection sampling caused by the vanilla IS policy gradient, and provide theoretical guarantees about its bias and variance.

1. Introduction

For the scalability and latency of automated decision-making systems (e.g., search, recommendation, and retrieval augmented generation (RAG)), two-stage decision models are often used in real-life scenarios. For instance, to handle millions or billions of items in search and recommendation, simple models (e.g., two-tower models (Chen et al., 2019)) are often used to threshold the items to obtain hundreds or thousands of top-k candidates. Then, a more complex model generates the ranked list of items or identifies a single item to recommend to the user for improving the users' interaction signals (e.g., view time, total price of purchase) (Ma et al., 2020). Moreover, in RAG, the first stage model first retrieves relevant and informative documents from a large database, and then the second stage large language model (LLM) generates the sentence response to the given prompts based on the retrieved top-k documents (Lewis et al., 2020). In such applications, improving the quality of the ***first-stage candidate retrieval model*** is indispensable, as the ***second-stage output generation model*** can choose or generate output only from the retrieved candidates, affecting the performance of the ***overall (joint) decision policy*** significantly.

A dominant way of training a (first-stage) candidate retrieval model is to train a simple model using prediction loss, which is referred to as the collaborative filtering (CF) approach. This approach aims to regress the users' rating (explicit feedback) or clicks or viewtime (implicit feedback) with the predicted affinity score between users and items. While CF makes it easy to retrieve top-k document based on the affinity score, a shortcoming of this approach is that the logged data is often biased (e.g., users are more likely to rate their favorite items and the exposure of item is determined by the logging policy), which leads to inaccuracy of regression (Schnabel et al., 2016). Moreover, we have a mismatch in the objective in minimizing the prediction error and maximizing the users' response signals (i.e., reward), resulting in a sub-optimal policy (Chen et al., 2019; Ma et al., 2020). In response, off-policy learning (OPL) approaches (Chen et al., 2019; Ma et al., 2020) aim to derive an unbiased policy gradient to directly maximize the reward signal. However, this approach is also known to suffer from a high variance and deficient support when the original item pools are large (Saito & Joachims, 2022; Sachdeva et al., 2020). Note that, while both CF and OPL approaches are model-agnostic, a simple two-tower model, which first encodes user and item information separately into an embedding (Chen et al., 2019), is often used for its scalability.

The common limitation of these prevalent approaches is that users are often represented using a single preference

[*]Equal contribution [1]Department of Computer Science, Cornell University, Ithaca, NY. Correspondence to: Haruka Kiyohara <hk844@cornell.edu>.

SIM workshop in the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

vector, regardless of the model architecture (Guo et al., 2021). This results in skewed item distributions in the top-k candidate set, and only similar items (e.g., all items are in the "action" category of the movie) can be observed in the top-k list, leading to the *diversity* issues in the candidate set (Peng et al., 2024; Guo et al., 2021). However, diversity is indeed inherently favored by users in the following real-life situations.

Example 1 (News recommendation)**.** *Even when users have a favorite topic of news articles (e.g., sports), having diverse topics (e.g., sports, economics, and business) on the top page may be more attractive (e.g., increase their viewtime) than filling out all the articles from their favorite category.*

Example 2 (Opinion summarization)**.** *When writing product reviews, referring to multiple viewpoints from diverse documents may improve the depth of analysis than referring to similar documents.*

Existing approaches for introducing diversity in the candidate set are to explicitly enforce item diversification and calibration via post-processing (Antikacioglu & Ravi, 2017; Steck, 2018). However, these methods rely either on the hand-engineered hyperparameter to determine the degree of diversification (Antikacioglu & Ravi, 2017) or the original data distribution, which is biased towards the logging distribution (Steck, 2018). Unfortunately, *there is no existing work that automatically adjusts the degree of diversification and calibration in a way to maximize our objective function (e.g., viewtime or purchase amount).*

To circumvent the issue, **this paper studies an OPL approach that can learn *optimal degree of diversity*, in a way to maximize the users' response signals**. We achieve this goal by the following procedure. First, we introduce a conditional sampling framework for sampling multiple user preferences and the corresponding items, and provide a theoretical guarantee that the sampling process simulates the empirical distribution of a Mixture-of-Gaussian of user preferences. This shows that our model is more flexible than the previous approaches and enables us to sample diverse items as the top-k candidate set, without concentrating on similar items. Then, we also derive an (off-)policy gradient estimator under this framework and show that the estimated (off-)policy gradient is unbiased. We further identify the potential challenge of this vanilla (unbiased) policy gradient, including **rejection sampling, high variance, and deficient support**, and present an improved policy gradient estimator called **the kernel importance sampling (Kernel IS) estimator**. The key idea of Kernel IS is to share reward observations among similar outputs generated by the joint policy (e.g., in the two-stage process). Our theoretical analysis shows that Kernel IS can achieve small bias and variance simultaneously under a more relaxed condition of support than the vanilla estimator.

2. Preliminalies

We formulate the two-stage decision process (e.g., two-stage recommender systems and retrieval augmented generation; RAG) as a contextual bandit problem.

Let $x \in \mathcal{X} \in \mathbb{R}^{d_x}$ be a d_x-dimensional context (e.g., user profile or ID) generated from an unknown distribution $p(x)$. Let $y \in \mathcal{Y}$ be an output of a generative model (e.g., LLM (Lewis et al., 2020), recommender model (Ma et al., 2020), and item-ranker in search (Chen et al., 2019; Gao et al., 2023)). The output y is either discrete or continuous and may be in the form of texts or visuals. We consider the following two-stage decision policy π for generating the output y given context x:

$$\pi(y|x) := \sum_{\mathcal{A}^k} \pi^{(1)}(\mathcal{A}^k|x)\pi^{(2)}(y|x, \mathcal{A}^k)$$

where $\pi(y|x)$ is the probability of generating the output y given context x. The policy is decomposed to the first-stage ($\pi^{(1)}$) and the second-stage ($\pi^{(2)}$) policies. $\mathcal{A}^k := (a_1, a_2, \cdots, a_j, \cdots, a_k)$ is referred to the top-k candidate set, where each a_j is in the (large) item pool of $\mathcal{A}$ and there are no duplicates in the candidate set (i.e., $a_j \neq a_l, \forall 1 \leq j < l \leq k$). In a large-scale recommender and search systems, $\mathcal{A}_k$ can be a (small) set of items and y can be the single item or ranking of items presented in the interface. In RAG, $\mathcal{A}_k$ can be the reference documents and y is the sentence response generated by LLM ($\pi^{(2)}$) using the documents ($\mathcal{A}_k$). In the rest of the paper, we consider the candidate retrieval policy ($\pi^{(1)}$) and assume that a static late-stage generation model ($\pi^{(2)}$) is given.

Once we generate the output y using the two-stage decision policy, users will respond with a reward $r \in \mathbb{R}$ (e.g., purchases or view time) following an unknown distribution $p(r|x, y)$. Our goal is to maximize the *policy value* defined as the expected reward under the given policy π,

$$V(\pi) := \mathbb{E}_{p(x)\pi(y|x)p(r|x,y)}[r],$$

using the logged data collected by some *logging policy* π_0 ($\neq \pi$) in the system's past operation:

$$\mathcal{D} := \{(x_i, y_i, r_i)\}_{i=1}^n \sim \prod_{i=1}^n p(x)\pi_0(y|x)p(r|x, y)$$

Note that the logging policy π_0 should not necessarily be a two-stage decision policy (i.e., can be a single stage one). Below, we use $q(x, y) = \mathbb{E}[r|x, y]$ to denote the expected reward.

2.1. Conventional approaches

In the candidate retrieval process where the policy needs to handle millions or billions of items, **two-tower models**

are often employed due to their computation efficiency (He et al., 2017; Chen et al., 2019; Ma et al., 2020). Two-tower models calculate the context-action logit as $\langle \mu(x), \phi(a) \rangle$, where $\mu(x)$ is the context encoding into a *"preference"* vector and $\phi(a)$ is the item embeddings. $\langle \cdot, \cdot \rangle$ denotes the inner product between the two inputs.

Under the above formulation, existing work is roughly categorized into two approaches for training the preference model (μ)[1]. The first one is **Collaborative-Filtering (CF)**, which aims to estimate the affinity score between contexts and items using reward signals as follows (He et al., 2016).

$$\ell(\mu; \phi) \approx \frac{1}{n} \sum_{i=1}^{n} (r_i - \langle \mu(x_i), \phi(a_i) \rangle)^2,$$

where we use the mean-squared error (MSE) loss as a standard choice of the loss function. Then, after training μ as the prediction model, we determine the candidate set by retrieving the top-k relevant items as follows.

$$w = \mu(x), \quad a_j = \mathrm{argmax}_{a \in \mathcal{A} \setminus \mathcal{A}^{(j-1)}} \langle w, \phi(a) \rangle,$$

The benefit of this approach is to minimize the computational cost by reducing the problem to regression. However, the disadvantage is that the loss function does not align with the objective function (i.e., maximizing the policy value), and inaccurate predictions for unobserved context-action pair (x, a) can easily lead to a sub-optimal policy (Schnabel et al., 2016; Ma et al., 2020).

To align the loss function to the objective of maximizing the policy value, **Off-Policy Learning (OPL)** aims to optimize the preference model by propagating (off-)policy gradient (Ma et al., 2020):

$$\nabla_\mu \hat{V}(\pi) \approx \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\pi^{(1)}} \left[\frac{\pi^{(2)}(y_i|x_i, \mathcal{A}^k)}{\pi_0(y_i|x_i)} \nabla_\mu \log \pi^{(1)}(\mathcal{A}^k|x_i) \right] r_i. \tag{1}$$

Note that the outer expectation $\mathbb{E}_{\pi^{(1)}}[\cdot]$ can be simulated by sampling a candidate set $\mathcal{A}^k$ using the candidate selection policy $\pi^{(1)}$. Plackett-luce (Guiver & Snelson, 2009) is considered the standard modeling of the candidate selection policy, as it enables conditionally independent sampling of k items without replacement as follows.

$$w = \mu(x), \quad a_j \sim \mathrm{softmax}_{a \in \mathcal{A} \setminus \mathcal{A}^{(j-1)}} (\langle w, \phi(a) \rangle / \gamma) \tag{2}$$

where the softmax operation is $\mathrm{softmax}_{z \in \mathcal{Z}}(z/\gamma) := \exp(z/\gamma)/(\sum_{z' \in \mathcal{Z}} \exp(z/\gamma))$ and $\gamma \in \mathbb{R}^{\neq 0}$ is its temperature parameter. Eq. (2) means that the Plackett-luce policy recursively applies softmax on the remaining items ($\mathcal{A} \setminus \mathcal{A}^{(j-1)}$) until we sample k different items as $\mathcal{A}^k$. The benefit of Plackett-luce is that the log likelihood $\nabla_\mu \log \pi^{(1)}(\mathcal{A}^k|x_i)$ is decomposed into $\sum_{j=1}^{k} \nabla_\mu \log \pi^{(1)}(a_j|x_i, \mathcal{A}^{(j-1)})$. The policy gradient defined in Eq. (1) enables unbiased estimation when the following support condition is satisfied: $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$, $\pi(y|x) > 0 \implies \pi_0(y|x) > 0$. However, OPL suffers from rejection sampling (i.e., gradient becomes zero as y_i cannot be generated from $\mathcal{A}^k$), high variance, and deficient support issues when the item pool is large (Sachdeva et al., 2020; Saito et al., 2024; Kiyohara et al., 2025).

Common failure mode of existing approaches. In addition to the aforementioned limitations, the common shortcoming of the existing approaches is to assume that user preference can be represented as a single vector (Guo et al., 2021). Specifically, a single representation of user preference can result in the concentration of items to some specific categories in the movie recommendation, failing to provide a diverse set of items as the candidate set. This is problematic in situations where the diversity of items is inherently valued – e.g., in news recommendation platforms, presenting articles from multiple topics (e.g., sports, economics, and politics) should increase the total view time compared to presenting a single topic (e.g., sports), even when the user has a favorite topic (e.g., sports). This motivates us a more flexible framework than existing works that can retrieve a diverse set of items ($\mathcal{A}^k$) in the candidate retrieval process to maximize the users' engagement signals (i.e., rewards).

3. Sampling diverse items in the candidate sets

Our key idea for selecting *diverse* items in the candidate set ($\mathcal{A}^k$) is to model the multi-modal distribution of user preference and sample actions from multiple user preference vectors. Specifically, we consider the following iterative sampling procedure of preference weight w_j and item a_j to sample items based on different preferences for each j:

$$\begin{aligned} w_j &\sim \mu_j(w|x, \mathcal{A}^{(j-1)}), \\ a_j &\sim \mathrm{softmax}_{a \in \mathcal{A} \setminus \mathcal{A}^{(j-1)}} (\langle w_j, \phi(a) \rangle / \gamma), \end{aligned} \tag{3}$$

This is a generalized framework that includes existing work (Ma et al., 2020) in its special case. We can parametrize $\mu_j(w|x, \mathcal{A}^{(j-1)})$ using a neural network (NN), taking the previous items as input. For example, when μ_j follows a normal distribution, NN outputs the mean vector and covariance matrix and w_j follows the corresponding normal distribution. We summarize conceptual differences between ours and existing works in Figure 1 and provide a theorem to show that the proposed framework simulates the multi-modal distribution of user preference in Appendix C.

[1]While we jointly optimize the context encoding (μ) and item embeddings (ϕ), this paper takes a closer look at the user preference model (μ) to consider diversity in the candidates.

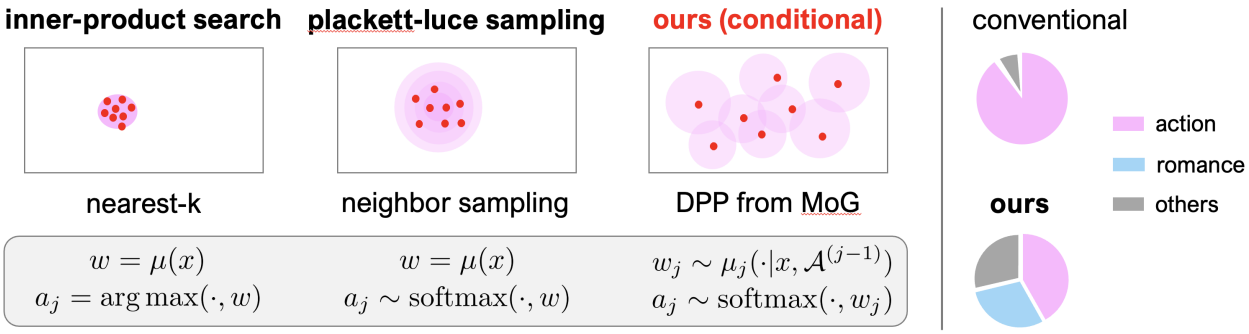

Figure 1. **(Left) Conceptual comparison between the proposed method and conventional approaches and (Right) the resulting proportions of categories in candidates ($\mathcal{A}^k$).** While the baselines represent a single preference per context, our sampling process simulates a more complex, Determinantal Point Process (DPP) sampling from a mixture-of-Gaussian (MoG) distribution. This helps model users' multiple and distributional interests such as preferring action movies for 45% of time and romance movies for 30% of time.

3.1. Data Efficient Policy Gradient Estimation

Next, we derive a (off-)policy gradient estimator corresponding to the sampling procedure described in Eq. (3). We also address the **rejection sampling, high variance, and deficient support** issues by considering the marginal distribution of the outputs (y) as follows.

$$\nabla_\mu V(\pi) \approx \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{p(\mathcal{W}^k, \mathcal{A}^k|x_i)}\Bigg[\frac{\pi^{(2)}(\psi(y_i)|x_i, \mathcal{A}^k)}{\pi_0(\psi(y_i)|x_i)} \nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x_i)\Bigg] r_i, \quad (4)$$

where $p(\mathcal{W}^k, \mathcal{A}^k|x)$ is the joint probability of sampling $\mathcal{W}^k$ and $\mathcal{A}^k$ given context x.[2] $\psi(y)$ is the marginal distribution of the output y. Following Kiyohara et al. (2025), we define the marginal distribution using a kernel function:

$$\pi(\psi(y)|z) = \int_{y' \in \mathcal{Y}} K(y, y'; z, \tau)\pi(y'|z)dy' = \mathbb{E}_{\pi(y|z)}[K(y, y'; z, \tau)] \quad (5)$$

where $z := (x, \mathcal{A}^k)$ are some the conditioning variables. A kernel function $K(\cdot, \cdot)$ should satisfy $\int_{y' \in \mathcal{Y}} K(y, y'; z, \tau)\mathrm{d}y' = 1, \forall (z, y) \in \mathcal{Z} \times \mathcal{Y}$, and τ is a bandwidth hyperparameter. A popular choice of kernel is a Gaussian kernel, which distributes kernel weights proportional to the similarly (i.e., embedding distance $d(y, y')$) is between two outputs as $K(y', y;\ z, \tau) \propto \exp(-d(y, y'))$. Using the kernel, we can re-write the gradient estimation (Eq. (10)) as follows.

$$\nabla_\mu V(\pi) \approx \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{p(\mathcal{W}^k, \mathcal{A}^k|x_i)\pi^{(2)}(y|x_i, \mathcal{A}^k(,\mathcal{W}^k))}\Bigg[\quad (6)$$

$$\frac{K(y, y_i; x_i, \tau)}{\pi_0(\psi(y_i)|x_i)} \nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x_i)\, r_i\Bigg], \quad (7)$$

The key points of the above **kernel importance sampling (Kernel IS)** estimator are the two-folds: (1) Kernel IS applies **soft rejection sampling**, by sampling output y using the late-stage policy $\pi^{(2)}$ and multiplying the kernel weight $K(y, y_i; x_i, \tau)$. This avoid hard rejection sampling in the vanilla IS, avoiding the zero-gradient. (2) Kernel IS also use **marginal density of the logging policy** ($\pi_0(\psi(y_i)|x_i)$) instead of the exact propensity ($\pi_0(y_i|x_i)$), where we provide the procedure to estimate the logging marginal density in Appendix B. The use of logging marginal density also helps mitigate high variance and deficient support. Together, Kernel IS has the following favorable statistical properties.

- Kernel IS relaxes the condition for support compared to the vanilla IS, mitigating the deficient support issue.
- Bias of Kernel IS can be small when within-kernel distribution shift between π and π_0 is small. This is achieved when using a smooth kernel like a Gaussian kernel or a small bandwidth hyperparameter τ.
- Kernel IS reduces the variance of the vanilla IS, and variance reduction becomes large when kernel bandwidth τ is large.

We provide the detailed discussion and proofs in Appendix C and D. For future work, we plan to conduct synthetic and real-data experiments and empirically compare the proposed method with the baselines.

[2] Because we maintain the conditional independence in the sampling process, $\nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x_i)$ is equivalent to the sum of $\nabla_{w_j} \log \pi^{(1)}(a_j|x_i, w_j, \mathcal{A}^{(j-1)})$ and $\nabla_{\mu_j} \log \mu_j(w_j|x_i, \mathcal{A}^{(j-1)})$.

References

Antikacioglu, A. and Ravi, R. Post processing recommender systems for diversity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 707–716, 2017.

Beygelzimer, A. and Langford, J. The offset tree for learning with partial labels. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 129–138, 2009.

Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., and Chi, E. H. Top-k off-policy correction for a reinforce recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 456–464, 2019.

Dudík, M., Langford, J., and Li, L. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 1097–1104, 2011.

Gao, G., Chang, J. D., Cardie, C., Brantley, K., and Joachim, T. Policy-gradient training of language models for ranking. *arXiv preprint arXiv:2310.04407*, 2023.

Guiver, J. and Snelson, E. Bayesian inference for plackett-luce ranking models. In *Proceedings of the 26th International Conference on Machine Learning*, pp. 377–384, 2009.

Guo, W., Krauth, K., Jordan, M., and Garg, N. The stereotyping problem in collaboratively filtered recommender systems. In *Proceedings of the 1st ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, pp. 1–10, 2021.

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

He, X., Zhang, H., Kan, M.-Y., and Chua, T.-S. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 549–558, 2016.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182, 2017.

Jiang, Z., Xu, F. F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J., Yang, Y., Callan, J., and Neubig, G. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, 2023.

Kallus, N. and Zhou, A. Policy evaluation and optimization with continuous treatments. In *International Conference on Artificial Intelligence and Statistics*, pp. 1243–1251, 2018.

Kiyohara, H., Saito, Y., Matsuhiro, T., Narita, Y., Shimizu, N., and Yamamoto, Y. Doubly robust off-policy evaluation for ranking policies under the cascade behavior model. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, pp. 487–497, 2022.

Kiyohara, H., Uehara, M., Narita, Y., Shimizu, N., Yamamoto, Y., and Saito, Y. Off-policy evaluation of ranking policies under diverse user behavior. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1154–1163, 2023.

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

Kiyohara, H., Cao, D. Y., Saito, Y., and Joachims, T. Prompt optimization with logged bandit data. *arXiv preprint arXiv:2504.02646*, 2025.

Konda, V. and Tsitsiklis, J. Actor-critic algorithms. *Advances in Neural Information Processing Systems*, 12, 1999.

Kulesza, A., Taskar, B., et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

Ma, J., Zhao, Z., Yi, X., Yang, J., Chen, M., Tang, J., Hong, L., and Chi, E. H. Off-policy learning in two-stage recommender systems. In *Proceedings of The Web Conference 2020*, pp. 463–473, 2020.

McInerney, J., Brost, B., Chandar, P., Mehrotra, R., and Carterette, B. Counterfactual evaluation of slate recommendations with sequential reward interactions. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1779–1788, 2020.

Metelli, A. M., Russo, A., and Restelli, M. Subgaussian and differentiable importance sampling for off-policy evaluation and learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Peng, K., Raghavan, M., Pierson, E., Kleinberg, J., and Garg, N. Reconciling the accuracy-diversity trade-off in recommendations. In *Proceedings of the ACM Web Conference 2024*, pp. 1318–1329, 2024.

Precup, D., Sutton, R. S., and Singh, S. P. Eligibility traces for off-policy policy evaluation. In *Proceedings of the 17th International Conference on Machine Learning*, pp. 759–766, 2000.

Sachdeva, N., Su, Y., and Joachims, T. Off-policy bandits with deficient support. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 965–975, 2020.

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.

Saito, Y. and Joachims, T. Off-policy evaluation for large action spaces via embeddings. In *Proceedings of the 39th International; Conference of Machine Learning*, volume 162, pp. 19089–19122, 2022.

Saito, Y., Ren, Q., and Joachims, T. Off-policy evaluation for large action spaces via conjunct effect modeling. In *Proceedings of the 40th International; Conference of Machine Learning*, volume 202, pp. 29734–29759, 2023.

Saito, Y., Yao, J., and Joachims, T. Potec: Off-policy learning for large action spaces via two-stage policy decomposition. *arXiv preprint arXiv:2402.06151*, 2024.

Sakhi, O., Aouali, I., Alquier, P., and Chopin, N. Logarithmic smoothing for pessimistic off-policy evaluation, selection and learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 80706–80755, 2024.

Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., and Joachims, T. Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1670–1679. PMLR, 2016.

Shimizu, T., Tanaka, K., Kishimoto, R., Kiyohara, H., Nomura, M., and Saito, Y. Effective off-policy evaluation and learning in contextual combinatorial bandits. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pp. 733–741, 2024.

Steck, H. Calibrated recommendations. In *Proceedings of the 12th ACM conference on Recommender Systems*, pp. 154–162, 2018.

Strehl, A., Langford, J., Li, L., and Kakade, S. M. Learning from logged implicit exploration data. In *Advances in Neural Information Processing Systems*, volume 23, pp. 2217–2225, 2010.

Su, Y., Dimakopoulou, M., Krishnamurthy, A., and Dudík, M. Doubly robust off-policy evaluation with shrinkage. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 9167–9176. PMLR, 2020.

Swaminathan, A., Krishnamurthy, A., Agarwal, A., Dudik, M., Langford, J., Jose, D., and Zitouni, I. Off-policy evaluation for slate recommendation. In *Advances in Neural Information Processing Systems*, volume 30, pp. 3632–3642, 2017.

Vlassis, N., Chandrashekar, A., Gil, F. A., and Kallus, N. Control variates for slate off-policy evaluation. *arXiv preprint arXiv:2106.07914*, 2021.

Wang, Y.-X., Agarwal, A., and Dudik, M. Optimal and Adaptive Off-policy Evaluation in Contextual Bandits. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3589–3597, 2017.

Yan, S.-Q., Gu, J.-C., Zhu, Y., and Ling, Z.-H. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

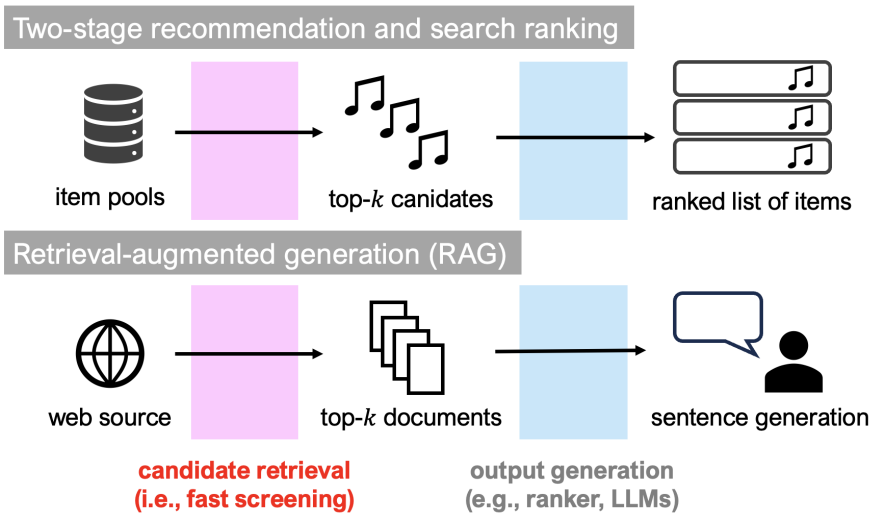

Figure 2. **Examples of two-stage decision systems.** (Top) Large-scale recommender and search systems and (Bottom) retrieval-augmented generation (RAG) with large language models (LLMs).

A. Related Work

We summarize the most notable related work.

A.1. Candidate selection for two-stage recommender and search systems

Computational efficiency is one of the key requirement in the candidate retrieval process where we need to process millions or billions of items. For this reason, two-tower models, which encodes user and item features separately, are often used, and there are two prevalent approaches in training this model. The first and the most traditional approach is collaborative filtering (CF) (He et al., 2017). This approach aims to enable accurate prediction of the affinity score between user-item pairs. However, because the objective of the CF-based training is minimizing estimation error, this approach has objective mismatch to the objective of maximizing the users' feedback signal (i.e., policy value). In contrast, the second approach called off-policy learning (OPL), calculates the (off-)policy gradient using logged data and update the model directly by the policy gradient (Chen et al., 2019; Ma et al., 2020). In particular, (Ma et al., 2020) enables unbiased estimation of the gradient of the candidate retrieval policy. However, both CF-based and OPL-based approaches shares the same limitation about *diversity* in the candidate set (Guo et al., 2021; Peng et al., 2024). In particular, (Guo et al., 2021) empirically demonstrate that the top-k candidates often concentrate on similar items (e.g., same "action" category) on a movie recommendation benchmark (Harper & Konstan, 2015) when using a single user representation. Existing approaches to dealing with such issues are to apply post-processing diversification (Antikacioglu & Ravi, 2017) or calibration (Steck, 2018) to include multiple categories of items in the top-k list. However, such calibration is often based on the biased logged data, and the degree of diversification depends on the hand-engineered hyperparameters or biased logged data. In contrast, our policy gradient approach identifies the *optimal* calibration or diversification of the candidate set in a way to maximize the downstream task reward (i.e., reward obtained by the given second-stage policy).

A.2. Retrieval-Augmented Generation

As another important application of the two-stage decisions, retrieval augmented generation (RAG) first retrieves the top-k document from a large web source and then generates a sentence-response using large language models (LLMs). RAG is originally designed for "accuracy"-seeking tasks like question and answering (QA), and therefore, retrieving the k most relevant items was considered the optimal solution (Lewis et al., 2020; Yan et al., 2024). For example, Lewis et al. (2020) propagate the task-specific loss, such as the classification loss in the QA task through the (second-stage) generation and (first-stage) retrieval models in an end-to-end manner. Jiang et al. (2023) gradually updates the retrieved document throughout the sentence generation phase, employing a similar top-k retrieval as Lewis et al. (2020). Our work differs from

the existing works in two ways. First, in applications like a news summary, retrieving diverse opinions on the news can be important for generating an in-depth technical report. However, retrieving items focuses on the "relevance" as existing work does (Yan et al., 2024) may collect only similar opinions, and ours enables us to collect a more diverse set of top-k documents. Moreover, the existing work does not consider an off-policy setting where the user response signals called *implicit feedback* (e.g., clicks or viewtime) are collected by the logging policy. Our work is the first to consider the OPL of a retrieval-augmentation policy.

A.3. Data-efficient Off-Policy Evaluation and Learning

Off-policy evaluation and learning (OPE/L) aims to evaluate or learn a new policy using logged data collected in the past operation of the system. Most OPE/L papers focus on the evaluation and learning of the second-stage generation policy ($\pi^{(2)}(y|x, \mathcal{A}^k)$) (Wang et al., 2017; Su et al., 2020; Metelli et al., 2021; Saito & Joachims, 2022; Sakhi et al., 2024). Among them, regression-based (Konda & Tsitsiklis, 1999; Beygelzimer & Langford, 2009), importance sampling (IS)-based (Precup et al., 2000; Strehl et al., 2010), and doubly robust (DR) (Dudík et al., 2011; Saito et al., 2024) approaches are three prevalent methods. The regression-based approach first trains a regression model to estimate the expected reward given context (x) and output (y). Then, it uses the imputed reward to estimate the policy value and gradient. While this approach enables low-variance estimation, OPE/L often suffers from high bias when the regression is inaccurate, which is common in OPE/L due to the skewed distribution of logged data. In contrast, IS reweights the reward observation in the logged data to enable unbiased estimation of the policy value and gradient. DR and its variants combine regression and IS to improve data efficiency compared to vanilla IS, while maintaining unbiasedness under certain conditions. However, both IS and DR are known to suffer from high variance and deficient support when the action space is large (Sachdeva et al., 2020; Saito & Joachims, 2022). To deal with these issues, some OPE/L work introduces an improved support condition and data efficiency using auxiliary action embeddings (Saito & Joachims, 2022; Saito et al., 2023; 2024; Sachdeva et al., 2024) or by leveraging similarity among items using kernels (Kallus & Zhou, 2018; Kiyohara et al., 2025). In particular, Kiyohara et al. (2025) considers the OPL of a prompt policy for personalized sentence generation and develops a method to estimate the gradient of the prompt policy leveraging similarity among generated sentences using kernel IS. Our work is inspired by this approach and proposed a generalized framework for applying kernel IS for data-efficient OPL for candidate retrieval.

While the OPE/L literature is sparse for the candidate retrieval policy ($\pi^{(1)}(\mathcal{A}^k|x)$), Chen et al. (2019) and Ma et al. (2020) are relevant to ours. Chen et al. (2019) calculates the gradient solely based on the probability that each item a is selected among the top-k (i.e., $\mathcal{A}^k$). This gradient is biased, as it does not refer to the output generation probability of the second-stage policy. In contrast, Ma et al. (2020) estimates the unbiased estimation by taking the second-stage policy into account. However, as discussed in Sections 2.1 and B.1, two sets of issues remain: (1) anti-diversity in the top-k candidates and (2) rejection sampling, high variance, and deficient support issues. In response, we dealt with the first problem by introducing a more generalized and flexible action sampling framework in Section 3 and deriving an unbiased policy gradient under our new framework. This *vanilla* policy gradient coincides with Ma et al. (2020) in its special case. Moreover, to mitigate the second problem, we also presented the kernel IS estimator and demonstrated its effectiveness in the experiments. Note that while we also have several related works on OPE/L of ranking (i.e., presenting items from the ranking order) (McInerney et al., 2020; Kiyohara et al., 2022; 2023) and other combinatorial actions (Swaminathan et al., 2017; Vlassis et al., 2021; Kiyohara et al., 2024; Shimizu et al., 2024), these methods also suffer from high variance when the item pool is large. Our method enables tractable and data-efficient OPE/L for the candidate retrieval process for the first time by leveraging the similarity among items.

B. Motivation of the proposed policy gradient method

Here, we provide the detailed step for deriving the proposed policy gradient.

B.1. Vanilla Policy Gradient

We first derive the *vanilla importance sampling (IS)* policy gradient following the sampling procedure described in Eq. (3) as follows.

$$\nabla_\mu V(\pi) \approx \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{p(\mathcal{W}^k, \mathcal{A}^k|x_i)} \left[\frac{\pi^{(2)}(y_i|x_i, \mathcal{A}^k)}{\pi_0(y_i|x_i)} \nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x_i) \right] r_i, \tag{8}$$

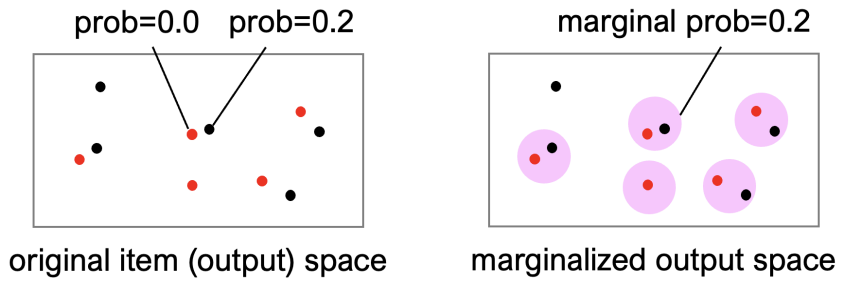

Figure 3. **Example of mitigating the deficient support issue (Sachdeva et al., 2020) using kernel smoothing in single item recommendation.** We consider the case where the logging candidate retrieval model is deterministic and red dots correspond to the supported items. The probability is based on the joint policy, i.e., $\pi(y|x)$. The actual marginalization with a Gaussian kernel results in a more gradual change of the marginal probability, but the right figure is simplified for clarity.

where $p(\mathcal{W}^k, \mathcal{A}^k)$ is the joint probability of sampling $\mathcal{W}^k$ and $\mathcal{A}^k$. Because we maintain the conditional independence in the sampling process, we have

$$\nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x_i) = \sum_{j=1}^{k} \left(\nabla_{w_j} \log \pi^{(1)}(a_j|x_i, w_j, \mathcal{A}^{(j-1)}) + \nabla_{\mu_j} \log \mu_j(w_j|x_i, \mathcal{A}^{(j-1)}) \right).$$

We provide the derivation in Appendix D.1. The benefit of this approach is that the estimated policy gradient is unbiased under the same support condition as conventional OPL methods (See Proposition 1 in Appendix C). However, the remaining issue that the gradient may suffer from high variance and deficient support when the output space ($\mathcal{Y}$) is large (e.g., recommender systems (Saito et al., 2024)) or high-dimensional (e.g., LLM (Kiyohara et al., 2025)). Moreover, in applications like single-item recommendations, the late-stage generation policy $\pi^{(2)}(y_i|x, \mathcal{A}^k)$ is often subject to rejection sampling if exactly the same item (y_i) is not included in $\mathcal{A}^k$, even when a similar item (y') is included in $\mathcal{A}^k$. This can be particularly problematic when we use a diverse set of candidates ($\mathcal{A}^k$) in our framework.

B.2. Data-efficient OPL via similarity-based smoothing with kernels

To deal with the issues of **high variance**, **deficient support**, and **rejection sampling**, we aim to **leverage similarity in the output** (y) **generated by the joint policy** π. We illustrate the motivative example in Figure 3. Specifically, consider a single-recommendation setting where the logging policy selects only partial items (e.g., based on top-k of CF). Because the vanilla policy gradient distinct each items regardless of the item similarity, the items not chosen by the logging policy suffers from the aforementioned issues even when we observe similar items.

Our central idea for tackling the problem is to distribute the reward observation among similar outputs, inspired by (Kiyohara et al., 2025). This becomes possible by considering the following marginalized distribution of output (y) as shown in Figure 3 (Right):

$$\pi(\psi(y)|x, z) = \int_{y' \in \mathcal{Y}} p(\psi(y)|x, y', z)\pi(y'|x, z) = \mathbb{E}_{\pi(y|x,z)}[p(\psi(y)|x, y', z)] \tag{9}$$

where z are some potential conditioning variables. When using a kernel function $K(\cdot, \cdot; \tau)$ with a bandwidth hyperparameter τ, we have $p(\psi(y)|x, y', z) = K(y, y'; x, \tau)$ where $\int_{Y' \in \mathcal{Y}} K(y, y'; x, \tau) = 1, \forall (x, y) \in \mathcal{X} \times \mathcal{Y}$. For instance, a Gaussian kernel distributes kernel weights proportional to the similarly (i.e., embedding distance $d(y, y')$) is between two outputs as $K(y', y;\ x, \tau) \propto \exp(-d(y, y'))$. Then, we define the ***Kernel IS*** policy gradient as

$$\nabla_\mu V(\pi) \approx \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{p(\mathcal{W}^k, \mathcal{A}^k|x_i)} \left[\frac{\pi^{(2)}(\psi(y_i)|x_i, \mathcal{A}^k)}{\pi_0(\psi(y_i)|x_i)} \nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x_i) \right] r_i \tag{10}$$

$$\approx \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{p(\mathcal{W}^k, \mathcal{A}^k|x_i)\pi^{(2)}(y|x_i, \mathcal{A}^k(, \mathcal{W}^k))} \left[\frac{K(y, y_i; x_i, \tau)}{\pi_0(\psi(y_i)|x_i)} \nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x_i)\, r_i \right] \tag{11}$$

From Eq. (10) to Eq. (11), we use Eq. (9) with z corresponding to $\mathcal{A}^k$, and the conditional independence of $(\mathcal{W}^k, \mathcal{A}^k, y)$ and (y_i, r_i) given x_i. The key points are the two-folds: (1) Kernel IS applies **soft rejection sampling**, by sampling output y

using the late-stage policy $\pi^{(2)}$ and multiplying the kernel weight $K(y, y_i; x_i, \tau)$. This avoid hard rejection sampling in the vanilla IS, avoiding the zero-gradient. (2) Kernel IS also use **marginal density of the logging policy** ($\pi_0(\psi(y_i)|x_i)$) instead of the exact propensity ($\pi_0(y_i|x_i)$). This also helps mitigate the high variance and deficient support issues.

While the precise computation of the expectation is costly, we can simulate the distribution using only one sample at each gradient step and repeat the process until the policy converges. The logging marginal density ($\pi_0(\psi(y)|x)$) can also be estimated via function approximation using the following loss (Kiyohara et al., 2025).

$$\ell(h; \pi_0, K, \tau) \approx \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\pi_0(y'|x_i)}[(h(x_i, y_i) - K(y_i, y'; x_i, \tau))^2].$$

where $h(\cdot)$ is the function approximation model. Because the expectation can be simulated by a single sample at each gradient step, the computation is tractable and does not scale with the size of output space ($\mathcal{Y}$). In Appendix C, we provide theoretical guarantees that the proposed estimator has favorable statistical properties in terms of bias and variance.

C. Theoretical analysis

We first show that the proposed conditional sampling process has the following favorable property about the candidate items ($\mathcal{A}^k$) and its distribution.

Theorem 1 (Determinantal Point Process (DPP) from a Mixture-of-Gaussian (MoG) distribution)**.** *When $\mu_j(w|x, \mathcal{A}^{(j-1)})$ follows a Gaussian distribution in Eq. (3), the preference set $\mathcal{W}_k := (w_1, w_2, \cdots, w_j, \cdots, w_k)$ can be seen as sampled from the DPP of MoG. That is, let $P(w|x)$ be some target distribution of $w \in \mathcal{W}$ following MoG. Then, the sampling probability is*

$$p(w_j|x, \mathcal{W}^{(j-1)}) \approx \frac{P(w|x)}{\widehat{P}(\mathcal{W}^{(j-1)}|x)}$$

where $\widehat{P}(\mathcal{W}^{(j-1)}|x)$ is the empirical distribution given the previous samples $\mathcal{W}^{(j-1)}$. The proof is provided in Appendix D.2.

Determinantal Point Process (DPP) is a sampling procedure that simulates a global distribution $P(w)$ using the empirical distribution of k points, i.e., $P(\mathcal{W}^k)$[3]. Therefore, Theorem 1 indicates that our sampling process simulates some global distribution of $P(w)$ using the sampled preference set $\mathcal{W}^k$ and the target distribution $P(w)$ can be a multi-modal distribution (e.g., Mixture-of-Gaussian). This enables the candidate selection policy to *calibrate* the item category based on the user preference (e.g., action movie is 45% and romance movie is 30%) to maximize the objective function (i.e., expected reward). This is a novel contribution of ours as there are no existing frameworks that consider the calibration and the objective maximization at the same time.

Then, we have the following proposition about the unbiasedness of the vanilla policy gradient (Eq. (8)) corresponding to the above sampling process.

Proposition 1 (Unbiasedness)**.** *The off-policy gradient ($\nabla_\mu \hat{V}(\pi)$) defined in the RHS of Eq. (8) is unbiased, i.e., $\mathbb{E}_{\mathcal{D}}[\nabla_\mu \hat{V}(\pi)] = \nabla_\mu V(\pi)$, when the support condition $\pi(y|x) > 0 \implies \pi_0(y|x) > 0, \forall (x, y) \in \mathcal{X} \times \mathcal{Y}$ is satisfied. The proof is provided in Appendix D.3.*

Next, we show that the Kernel IS keeps the bias small under relaxed condition about support. For this, we first introduce a new support condition called similar output support.

Definition 1 (Similar output support)**.** *The similar output support is satisfied if, $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$, $\pi(\psi(y)|x) > 0 \implies \pi_0(\psi(y)|x) > 0$.*

The similar output support condition is always satisfied when the original support condition is satisfied by the definition of Eq. (9). Moreover, this condition is always satisfied when using a smooth kernel like a Gaussian kernel. Under this relaxed condition, the bias of Kernel IS is characterized as follows.

[3]We can use a smaller preference set than candidate items, e.g., sampling only 10 preferences and sample 100 items, where each preference is used to sample 10 items if it adequately expresses the preference distribution.

Theorem 2 (Bias of the Kernel IS gradient estimator)**.** *Under the similar output support condition, the bias of the marginalized estimator is described as follows, under any choice of kernels and bandwidth hyperparameter.*

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}[\nabla_\mu \hat{V}(\pi)] - \nabla_\mu V(\pi) \\
&= \mathbb{E}_{p(x)\pi_0(y,\phi(y')|x)} \left[\mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x)} \left[\Delta_w(\psi(y),\psi(y');\mathcal{A}^k) \nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x) \right] q(x,y) \right] \\
&\quad + \mathbb{E}_{p(x)p(\mathcal{W}^k,\mathcal{A}^k|x)} \left[\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\, \Delta_q(\pi_0,\pi;\psi(y')) \right] \Big]
\end{aligned}$$

where $\Delta_w(\psi(y),\psi(y');\mathcal{A}^k)$ *refers to the difference of the importance weights* $w(x,\psi(y),\mathcal{A}^k) := \pi^{(2)}(\psi(y)|x,\mathcal{A}^k)/\pi_0(\psi(y)|x)$ *between* $\psi(y)$ *and* $\psi(y')$*.* $\Delta_q(\pi_0,\pi;\psi(y'))$ *refers to the difference of within-neighbor expected reward defined as* $q(\cdot,\pi_0) := \mathbb{E}_{\pi^{(2)}(\psi(y')|x,\mathcal{A}^k)}[q(x,\psi(y');\pi_0)]$ *between* π_0 *and* π*.* y *and* y' *should be similar samples within the kernel neighbor, as we consider the sampling process of* $\mathbb{E}_{\pi_0(y,\psi(y')|x)}[\cdot] = \mathbb{E}_{\pi_0(\psi(y')|x)\pi_0(y|x,\psi(y'))}[\cdot]$*. The proof is provided in Appendix D.4.*

In Theorem 2, the first term is often negligibly small when there are no abrupt changes of π, $\pi^{(2)}$, π_0 on $(x,\mathcal{A}^k,y)$, and the second term is the dominant term. The second term becomes small when the *within-neighbor expected reward difference* between π and π_0 is small. This condition is satisfied when using a small bandwidth hyperparameter τ or a smooth kernel like a Gaussian kernel.

In contrast, we have the following variance difference between the vanilla IS and Kernel IS.

Theorem 3 (Variance reduction of the kernel importance weight)**.** *The kernel IS estimator reduces the variance of the vanilla importance weight (conditioned on* $(x,\mathcal{W}^k,\mathcal{A}^k)$*) to the following degree.*

$$\mathbb{V}_y\left(\frac{\pi^{(2)}(y|x,\mathcal{A}^k)}{\pi_0(y|x)}\right) - \mathbb{V}_{\psi(y)}\left(\frac{\pi^{(2)}(\psi(y)|x,\mathcal{A}^k)}{\pi_0(\psi(y)|x)}\right) = \mathbb{E}_{\psi(y')}\left[\mathbb{V}_y\left(\frac{\pi^{(2)}(y|x,\mathcal{A}^k)}{\pi_0(y|x)}\right)\right].$$

The proof is provided in Appendix D.5.

Theorem 3 suggests that Kernel IS reduces the conditional variance caused within kernel neighbors. This is reasonable, as Kernel IS applies IS in the marginal output space and about IS for outputs within kernel neighbors. Therefore, we can expect a high variance reduction when the bandwidth hyperparameter τ is adequately large.

D. Omitted derivations and proofs

This section provides the derivations and proofs omitted in the previous sections.

D.1. Derivation of the policy gradients

We show the derivation of the original gradient (Eq. (8)).

$$
\begin{aligned}
\nabla_\mu \hat{V}(\pi) &\approx \frac{1}{n}\sum_{i=1}^{n} \frac{\nabla_\mu \pi(y_i|x_i)}{\pi_0(y_i|x_i)} r_i \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{\nabla_\mu \left(\sum_{(\mathcal{W}^k,\mathcal{A}^k)} \pi^{(2)}(y_i|x_i,\mathcal{W}^k,\mathcal{A}^k) p(\mathcal{W}^k,\mathcal{A}^k|x_i)\right)}{\pi_0(y_i|x_i)} r_i \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{\sum_{(\mathcal{W}^k,\mathcal{A}^k)} \pi^{(2)}(y_i|x_i,\mathcal{W}^k,\mathcal{A}^k) \nabla_\mu p(\mathcal{W}^k,\mathcal{A}^k|x_i)}{\pi_0(y_i|x_i)} r_i \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{\sum_{(\mathcal{W}^k,\mathcal{A}^k)} \pi^{(2)}(y_i|x_i,\mathcal{W}^k,\mathcal{A}^k)\, p(\mathcal{W}^k,\mathcal{A}^k|x_i)}{\pi_0(y_i|x_i)} \frac{\nabla_\mu p(\mathcal{W}^k,\mathcal{A}^k|x_i)}{p(\mathcal{W}^k,\mathcal{A}^k|x_i)} r_i \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{\sum_{(\mathcal{W}^k,\mathcal{A}^k)} \pi^{(2)}(y_i|x_i,\mathcal{W}^k,\mathcal{A}^k)\, p(\mathcal{W}^k,\mathcal{A}^k|x_i) \frac{\nabla_\mu p(\mathcal{W}^k,\mathcal{A}^k|x_i)}{p(\mathcal{W}^k,\mathcal{A}^k|x_i)}}{\pi_0(y_i|x_i)} r_i \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{\sum_{(\mathcal{W}^k,\mathcal{A}^k)} p(\mathcal{W}^k,\mathcal{A}^k|x_i)\, \pi^{(2)}(y_i|x_i,\mathcal{W}^k,\mathcal{A}^k) \nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x_i)}{\pi_0(y_i|x_i)} r_i \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x_i)} \left[\frac{\pi^{(2)}(y_i|x_i,\mathcal{W}^k,\mathcal{A}^k)}{\pi_0(y_i|x_i)} \nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x_i)\right] r_i.
\end{aligned}
$$

D.2. Proof of Theorem 1

Proof. We start by describing the general idea of the determinantal point process (DPP) (Kulesza et al., 2012). Let $P(z)$ be the original distribution of the variable z. Let $p(z_j|\mathcal{Z}^{(j-1)})$ be distribution of the j-th point. Also let $P(\mathcal{Z}^j)$ be joint distribution of j points and $\widehat{P}(\mathcal{Z}^j)$ be its empirical approximation. Then, we have the following relationship in the sampling process:

$$
P(\mathcal{Z}^j) = p(z_j|\mathcal{Z}^{(j-1)}) P(\mathcal{Z}^{(j-1)}). \tag{12}
$$

The goal of DPP is to simulate the original distribution $P(z)$ with the empirical distribution of j points, i.e., $P(z) \approx \widehat{P}(\mathcal{Z}^j)$. Therefore, sampling the j-th point follows the following distribution.

$$
p(z_j|\mathcal{Z}^{(j-1)}) \approx \frac{P(z)}{\widehat{P}(\mathcal{Z}^{(j-1)})}.
$$

This is how the DPP process works. Then, our goal is to show that the sampling from Eq. (3) simulates the DPP process, where $P(w|x)$ follows some mixture-of-gaussian (MoG) distribution. From Eq. (12), this can be immediately proved by defining $P(w|x) := \lim_{j\to\infty} p(w_j|x,\mathcal{W}^{(j-1)}) P(\mathcal{W}^{(j-1)}|x)$ and showing that $P(\mathcal{W}^{(j)}|x)$ follows MoG for all $j \in \mathbb{N}_{>1}$. We prove the latter statement by deduction. First, for $j = 1$, we know that $p(w_j|x)$ follows a single Gaussian distribution. Next, for $j > 1$ suppose that $p(\mathcal{W}^{(j-1)}|x)$ follows either a single Gaussian or MoG distribution. We also know that $p(w_j, |x, \mathcal{W}^{(j-1)}, \mathcal{A}^{(j-1)})$ follows Gaussian. Then, we have

$$
P(\mathcal{W}^j|x) = \underbrace{\left(\sum_{\mathcal{A}^{(j-1)}} p(w_j \,|\, x, \mathcal{W}^{(j-1)}, \mathcal{A}^{(j-1)}) P(\mathcal{A}^{(j-1)} \,|\, x, \mathcal{W}^{(j-1)})\right)}_{=MoG\ (1)} \cdot \underbrace{P(\mathcal{W}^{(j-1)} \,|\, x)}_{=Gaussian\ or\ MoG\ (2)},
$$

where $P(\mathcal{A}^{(j-1)} \,|\, x, \mathcal{W}^{(j-1)})$ is the weight of MoG (1). Because the product of the two Gaussian distributions results in a Gaussian distribution, the product (i.e., convolution) of two MoGs or that of a Gaussian and MoG results in another MoG. Thus, $p(\mathcal{W}_j|x)$ follows MoG. Therefore, we can see the sampling process in Eq. (3) as a DPP where the target distribution $P(w|x)$ is MoG. □

D.3. Proof of Proposition 1

Proof. We show that our gradient estimator enables an unbiased estimation of the policy gradient. In the proofs, we use $q(x, y) := \mathbb{E}[r|x, y]$ to denote the expected reward given context x and output y and $q(x, \mathcal{A}^k(, \mathcal{W}^k); \pi^{(2)}) := \mathbb{E}[r|x, \mathcal{A}^k(, \mathcal{W}^k); \pi^{(2)}]$ be the expected reward given context x and candidate set $\mathcal{A}^k$ under the given output generation policy $\pi^{(2)}$.

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{D}}[\nabla_\mu \hat{V}(\pi)] \\
&= \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{p(\mathcal{W}^k, \mathcal{A}^k|x)}\left[\frac{\pi^{(2)}(y|x, \mathcal{A}^k)}{\pi_0(y|x)} \nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x)\right] r\right] \\
&= \mathbb{E}_{p(x)\pi_0(y|x)p(r|x,y)}\left[\mathbb{E}_{p(\mathcal{W}^k, \mathcal{A}^k|x)}\left[\frac{\pi^{(2)}(y|x, \mathcal{A}^k)}{\pi_0(y|x)} \nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x)\right] r\right] \\
&= \mathbb{E}_{p(x)\pi_0(y|x)}\left[\mathbb{E}_{p(\mathcal{W}^k, \mathcal{A}^k|x)}\left[\frac{\pi^{(2)}(y|x, \mathcal{A}^k)}{\pi_0(y|x)} \nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x)\right] q(x, y)\right] \qquad (13) \\
&= \mathbb{E}_{p(x)\pi_0(y|x)}\left[\frac{1}{\pi_0(y|x)} \mathbb{E}_{p(\mathcal{W}^k, \mathcal{A}^k|x)}\left[\pi^{(2)}(y|x, \mathcal{A}^k) \nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x)\right] q(x, y)\right] \\
&= \mathbb{E}_{p(x)}\left[\int_{y \in \mathcal{Y}} \pi_0(y|x) \frac{1}{\pi_0(y|x)} \mathbb{E}_{p(\mathcal{W}^k, \mathcal{A}^k|x)}\left[\pi^{(2)}(y|x, \mathcal{A}^k) \nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x)\right] q(x, y)\, \mathrm{d}y\right] \\
&= \mathbb{E}_{p(x)}\left[\mathbb{E}_{p(\mathcal{W}^k, \mathcal{A}^k|x)}\left[\left(\int_{y \in \mathcal{Y}} \pi^{(2)}(y|x, \mathcal{A}^k(, \mathcal{W}^k)) q(x, y)\, \mathrm{d}y\right) \nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x)\right]\right] \qquad (14) \\
&= \mathbb{E}_{p(x)}\left[\mathbb{E}_{p(\mathcal{W}^k, \mathcal{A}^k|x)}\left[\nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x)\right] q(x, \mathcal{A}^k(, \mathcal{W}^k); \pi^{(2)})\right] \\
&= \mathbb{E}_{p(x)p(\mathcal{W}^k, \mathcal{A}^k|x)}\left[\nabla_\mu \log p(\mathcal{W}^k, \mathcal{A}^k|x)\, q(x, \mathcal{A}^k(, \mathcal{W}^k); \pi^{(2)})\right]. \qquad (15)
\end{aligned}
$$

□

D.4. Proof of Theorem 2

Proof. As a preparation, let $\Delta_w(\psi(y), \psi(y'); \mathcal{A}^k) = w(x, \psi(y), \mathcal{A}^k) - w(x, \psi(y'), \mathcal{A}^k)$ where

$$
w(x, \psi(y), \mathcal{A}^k) := \frac{\pi^{(2)}(\phi(y))|x, \mathcal{A}^k)}{\pi_0(\psi(y)|x)} = \frac{1}{\pi_0(\psi(y)|x)} \frac{p(\mathcal{A}^k|x, \psi(y); \pi^{(2)})\pi(\psi(y)|x)}{\pi^{(1)}(\mathcal{A}^k|x)}.
$$

Then we have

$$
\begin{aligned}
&\Delta_w(\psi(y), \psi(y'); \mathcal{A}^k) \\
&= \frac{1}{\pi^{(1)}(\mathcal{A}^k|x)}\left(\frac{\pi(\psi(y)|x)}{\pi_0(\psi(y)|x)} p(\mathcal{A}^k|x, \psi(y); \pi^{(2)}) - \frac{\pi(\psi(y')|x)}{\pi_0(\psi(y')|x)} p(\mathcal{A}^k|x, \psi(y'); \pi^{(2)})\right) \\
&= \frac{1}{\pi^{(1)}(\mathcal{A}^k|x)}\left(w(x, \psi(y)) p(\mathcal{A}^k|x, \psi(y); \pi^{(2)}) - w(x, \psi(y')) p(\mathcal{A}^k|x, \psi(y'); \pi^{(2)})\right).
\end{aligned}
$$

where $w(x, \psi(y)) := \pi(\psi(y)|x)/\pi_0(\psi(y)|x)$ is the distribution shift between π and π_0 in the marginalized outcome space. This suggests that if we do not have an abrupt change of π_0, $\pi^{(1)}$, and $\pi^{(2)}$ between the neighbors $\psi(y)$ and $\psi(y')$, $\Delta_w(\psi(y), \psi(y'))$ should be small.

Then, we derive the bias of the marginalized gradient estimator as follows. We use $q(x, y) := \mathbb{E}[r|x, y]$ to denote the expected reward given context x and output y and $q(x, \psi(y); \pi) := \mathbb{E}[r|x, \phi(y); \pi]$ be that given context x and marginalized neighbors $\psi(y)$ under the policy π.

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{D}}[\nabla_\mu \hat{V}(\pi)] \\
&= \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\frac{\pi^{(2)}(\psi(y)|x,\mathcal{A}^k)}{\pi_0(\psi(y)|x)}\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\right] r\right] \\
&= \mathbb{E}_{p(x)\pi_0(y|x)}\left[\mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\frac{\pi^{(2)}(\psi(y)|x,\mathcal{A}^k)}{\pi_0(\psi(y)|x)}\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\right] q(x,y)\right] \\
&= \mathbb{E}_{p(x)\pi_0(\phi(y')|x)\pi_0(y|x,\phi(y'))}\left[\mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\frac{\pi^{(2)}(\psi(y)|x,\mathcal{A}^k)}{\pi_0(\psi(y)|x)}\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\right] q(x,y)\right] \\
&= \mathbb{E}_{p(x)\pi_0(\phi(y')|x)\pi_0(y|x,\phi(y'))}\left[\mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\frac{\pi^{(2)}(\psi(y)|x,\mathcal{A}^k)}{\pi_0(\psi(y)|x)}\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\right] q(x,y)\right] \\
&\quad - \mathbb{E}_{p(x)\pi_0(\phi(y')|x)\pi_0(y|x,\phi(y'))}\left[\mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\frac{\pi^{(2)}(\psi(y')|x,\mathcal{A}^k)}{\pi_0(\psi(y')|x)}\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\right] q(x,y)\right] \\
&\quad + \mathbb{E}_{p(x)\pi_0(\phi(y')|x)\pi_0(y|x,\phi(y'))}\left[\mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\frac{\pi^{(2)}(\psi(y')|x,\mathcal{A}^k)}{\pi_0(\psi(y')|x)}\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\right] q(x,y)\right] \\
&= \mathbb{E}_{p(x)\pi_0(\phi(y')|x)\pi_0(y|x,\phi(y'))}\left[\mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\Delta_w(\psi(y),\psi(y');\mathcal{A}^k)\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\right] q(x,y)\right] \\
&\quad + \mathbb{E}_{p(x)\pi_0(\phi(y')|x)}\left[\mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\frac{\pi^{(2)}(\psi(y')|x,\mathcal{A}^k)}{\pi_0(\psi(y')|x)}\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\right]\int_{y\in\mathcal{Y}}\pi_0(y|x,\phi(y'))q(x,y)\,\mathrm{d}y\right] \\
&= \mathbb{E}_{p(x)\pi_0(\phi(y')|x)\pi_0(y|x,\phi(y'))}\left[\mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\Delta_w(\psi(y),\psi(y');\mathcal{A}^k)\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\right] q(x,y)\right] \\
&\quad + \mathbb{E}_{p(x)\pi_0(\phi(y')|x)}\left[\mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\frac{\pi^{(2)}(\psi(y')|x,\mathcal{A}^k)}{\pi_0(\psi(y')|x)}\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\right] q(x,\psi(y');\pi_0)\right].
\end{aligned}
$$

Next, we further decompose the second term. Because the last line corresponds to Eq. (13) in D.3 (the proof of Proposition 1; unbiasedness) with y corresponding to $\psi(y)$, we only show the lines corresponding to Eqs. (13), (14), and (15).

$$
\begin{aligned}
&\mathbb{E}_{p(x)\pi_0(\phi(y')|x)}\left[\mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\frac{\pi^{(2)}(\psi(y')|x,\mathcal{A}^k)}{\pi_0(\psi(y')|x)}\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\right] q(x,\psi(y');\pi_0)\right] \\
&= \mathbb{E}_{p(x)}\left[\mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\left(\int_{\psi(y)\in\Psi(\mathcal{Y})}\pi^{(2)}(\psi(y')|x,\mathcal{A}^k(,\mathcal{W}^k))q(x,\psi(y');\pi_0)\,\mathrm{d}\psi(y)\right)\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\right]\right] \\
&= \mathbb{E}_{p(x)p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\,\mathbb{E}_{\pi^{(2)}(\psi(y')|x,\mathcal{A}^k)}[q(x,\psi(y');\pi_0)]\right].
\end{aligned}
$$

The expected reward $\mathbb{E}_{\pi^{(2)}(\psi(y')|x,\mathcal{A}^k)}[q(x,\psi(y');\pi_0)]$ considers a situation where the probability of projecting output generation to $\psi(y)$ follows $\pi^{(2)}$, while when the aggregation of rewards within $\psi(y)$ follows π_0 (not π). This is because we correct the distribution shift of the marginalized distribution with the inverse propensity of $\pi_0(\psi(y)|x)$ but do not correct the within-neighbor distribution shift $\pi_0(y'|x,\psi(y))$ for the variance reduction purpose and to relax the support condition.

Finally, let $\Delta_q(\pi_0,\pi;\psi) := \mathbb{E}_{\pi^{(2)}(\psi(y')|x,\mathcal{A}^k)}[q(x,\psi(y');\pi_0)] - \mathbb{E}_{\pi^{(2)}(\psi(y')|x,\mathcal{A}^k)}[q(x,\psi(y');\pi)]$. The bias of the marginalized gradient estimator becomes as follows.

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{D}}[\nabla_\mu \hat{V}(\pi)] - \nabla_\mu V(\pi) \\
&= \mathbb{E}_{p(x)\pi_0(\phi(y')|x)\pi_0(y|x,\phi(y'))}\left[\mathbb{E}_{p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\Delta_w(\psi(y),\psi(y');\mathcal{A}^k)\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\right] q(x,y)\right] \\
&\quad + \mathbb{E}_{p(x)p(\mathcal{W}^k,\mathcal{A}^k|x)}\left[\nabla_\mu \log p(\mathcal{W}^k,\mathcal{A}^k|x)\,\Delta_q(\pi_0,\pi;\psi)\right].
\end{aligned}
$$

As discussed, the first term often becomes small when there are no abrupt changes in the policy in $(x,y)\in\mathcal{X}\times\mathcal{Y}$. The second term becomes small when using distance-based kernels such as a Gaussian kernel or when the bandwidth hyperparameter is small. □

D.5. Proof of Theorem 3

Proof. From the law of total variance, in general, we have

$$
\mathbb{V}_{Z|X}(f(Z)) = \mathbb{V}_{Y|X}(\mathbb{E}_{Z|X,Y}[f(Z)]) + \mathbb{E}_{Y|X}[\mathbb{V}_{Z|X,Y}(f(Z))|X].
$$

Therefore, if we show that $f(Y) = \mathbb{E}_{Z|X,Y}[f(Z)]$, we can say

$$\mathbb{V}_{Z|X}(f(Z)) - \mathbb{V}_{Y|X}(f(Y)) = \mathbb{E}_{Y|X}[\mathbb{V}_{Z|X,Y}(f(Z))|X].$$

Thus, we show that $w(x, \psi(y')) = \mathbb{E}_{\pi_0(y|x,\psi(y'))}[w(x, y)]$ to prove the reduction of the conditional variance, where we define

$$w(x, \psi(y)) := \frac{\pi^{(2)}(\psi(y)|x, \mathcal{A}^k)}{\pi_0(\psi(y)|x)}, \qquad w(x, y) := \frac{\pi^{(2)}(y|x, \mathcal{A}^k)}{\pi_0(y|x)}.$$

Then we have

$$\begin{aligned}
\mathbb{E}_{\pi_0(y|x,\psi(y'))}\left[\frac{\pi^{(2)}(y|x, \mathcal{A}^k)}{\pi_0(y|x)}\right] &= \int_{y\in\mathcal{Y}} \pi_0(y|x, \psi(y'))\frac{\pi^{(2)}(y|x, \mathcal{A}^k)}{\pi_0(y|x)}\,\mathrm{d}y \\
&= \int_{y\in\mathcal{Y}} \pi_0(y|x, \psi(y'))\frac{\pi^{(2)}(y|x, \mathcal{A}^k)}{\frac{\pi_0(y|x,\psi(y'))\pi_0(\psi(y')|x)}{p(\psi(y')|x,y)}}\,\mathrm{d}y \\
&= \int_{y\in\mathcal{Y}} \frac{\pi^{(2)}(y|x, \mathcal{A}^k)}{\pi_0(\psi(y')|x)} p(\psi(y')|x, y, (, \mathcal{A}^k))\mathrm{d}y \\
&= \frac{\pi^{(2)}(\psi(y')|x, \mathcal{A}^k)}{\pi_0(\psi(y')|x)}.
\end{aligned}$$

Therefore,

$$\mathbb{V}_y\left(\frac{\pi^{(2)}(y|x, \mathcal{A}^k)}{\pi_0(y|x)}\right) - \mathbb{V}_{\psi(y)}\left(\frac{\pi^{(2)}(\psi(y)|x, \mathcal{A}^k)}{\pi_0(\psi(y)|x)}\right) = \mathbb{E}_{\psi(y')}\left[\mathbb{V}_y\left(\frac{\pi^{(2)}(y|x, \mathcal{A}^k)}{\pi_0(y|x)}\right)\right].$$

□