GENERAL FRAMEWORK FOR OFF-POLICY LEARNING WITH PARTIALLY-OBSERVED REWARD

Anonymous authors
Paper under double-blind review

ABSTRACT

Off-policy learning (OPL) in contextual bandits aims to learn a decision-making policy that maximizes the target rewards by using only historical interaction data collected under previously developed policies. Unfortunately, when rewards are only partially observed, the effectiveness of OPL degrades severely. Well-known examples of such partial rewards include explicit ratings in content recommendations, conversion signals on e-commerce platforms that are partial due to delay, and the issue of censoring in medical problems. One possible solution to deal with such partial rewards is to use secondary rewards, such as dwelling time, clicks, and medical indicators, which are more densely observed. However, relying solely on such secondary rewards can also lead to poor policy learning since they may not align with the target reward. Thus, this work studies a new and general problem of OPL where the goal is to learn a policy that maximizes the expected target reward by leveraging densely observed secondary rewards as supplemental data. We then propose a new method called **Hy**brid Policy Optimization for **P**artially-Obs**e**rved **R**eward (**HyPeR**), which effectively uses the secondary rewards in addition to the partially-observed target reward to achieve effective OPL despite the challenging scenario. We also discuss a case where we aim to optimize not only the expected target reward but also the expected secondary rewards to some extent; counter-intuitively, we will show that leveraging the two objectives is in fact advantageous also for the optimization of only the target reward. Along with statistical analysis of our proposed methods, empirical evaluations on both synthetic and real-world data show that HyPeR outperforms existing methods in various scenarios.

1 INTRODUCTION

In applications that involve dynamic decision-making, such as ad placement and recommendation systems, exploration can be costly and risky. These constraints limit the use of online exploration of actions, thereby motivating the study of offline policy learning methods. **Off-policy learning (OPL)** addresses this need by enabling the policy optimization using only logged bandit data generated under old policies (Joachims et al., 2018; Su et al., 2019; 2020a; Uehara et al., 2022).

A typical approach to OPL is using policy gradient (PG), which we can estimate unbiasedly based on the logged data via techniques like Inverse Propensity Score (IPS) and Doubly Robust (DR) (Dudík et al., 2014). OPL methods based on PG iterations perform effectively in ideal scenarios with large sample sizes and relatively small action spaces (Saito & Joachims, 2021; Sachdeva et al., 2023; Taufiq et al., 2024). However, in many application scenarios, rewards are only partially observed due to missing data (Jakobsen et al., 2017; Wang et al., 2019b; Jadidinejad et al., 2019; Christakopoulou et al., 2022), censoring (Ren et al., 2019; Wang et al., 2019a), delayed observation (Wang et al., 2022b; Imbens et al., 2022; Wang et al., 2023; Saito et al., 2024), data fusion (Imbens et al., 2022), and multi-stage rewards (Wan & McAuley, 2018; Hadash et al., 2018; Ma et al., 2018; Saito, 2020). For instance, on e-commerce platforms, binary conversions serve as a target reward to maximize. Unfortunately, conversions are only observed for products that were clicked and seen by the user. To make matters worse, conversion signals are obtained only after weeks-long delays (Ktena et al., 2021), and most of them are yet to be observed when policy learning is conducted. Table 1 provides a list of real-life examples and causes of partially-observed rewards. When rewards are only partially observed due to such causes, the estimation of PG would result in high variance and inefficient OPL.

Table 1: Examples of Partially-Observed Rewards in Various Real-Life Scenarios

Scenario	**Reward**	**Reason for Partial Observation**
Streaming Platform	Explicit Rating	Multi-stage & Missing (Hadash et al., 2018)
E-commerce	Conversion	Multi-stage & Delayed (Ktena et al., 2021)
Ad-placement	Product Purchase	Missing & Delayed (Ashkan & Clarke, 2009)
Clinical Trials	Long-term Health	Censoring (Pratap et al., 2020)
Financial Investment	Long-term Return	Delayed (Arroyo et al., 2019)
Subscription Platform	User Retention	Delayed (Wang et al., 2022b)
Remediation	Academic Performance	Delayed (Barnett, 2011)

Our work proposes a new problem formulation to address this challenging but highly general problem: OPL with partial observations of the reward. In scenarios with only partial observations of the target reward, one possible idea to perform OPL more effectively is to use more frequently observed *secondary rewards* instead. For instance, in e-commerce recommender systems, we often observe not only the conversion signals but also other implicit feedback such as clicks and dwell time, which are much more densely observed with no missingness (Liu et al., 2010; Jadidinejad et al., 2019). Thus, in our problem formulation, we have access to both the *target reward* that is only partially observed (e.g., user ratings, conversions, retention, future earnings, and survival days) and the *secondary rewards* that are fully observed (e.g., clicks, dwell time, and short-term medical indicators). Considering the case where we also aim to optimize secondary rewards, we explore learning a policy that maximizes a weighted sum of the target reward and secondary rewards. This represents a more general objective rather than focusing only on the optimization of the target reward.

With the availability of the secondary rewards, one feasible approach to perform OPL is to maximize some aggregation of the secondary rewards as a surrogate for the target reward. Nonetheless, a significant drawback of this approach is the potential for high PG estimation bias, as secondary rewards often do not align perfectly with the target reward (Liu et al., 2010; Jadidinejad et al., 2019). We also do not know how to construct an appropriate aggregation function of the secondary rewards. Therefore, there is a dilemma between the target and secondary rewards: the former more accurately aligns with our ultimate objective, while the latter provides more observations. Using only the former leads to high variance in PG estimation due to missing observations, while relying solely on the latter results in high bias due to a potential misalignment with the ultimate objective.

To solve this new OPL problem more effectively, we focus on developing a method that can leverage both target and secondary rewards in a principled way. Specifically, we propose **Hy**brid Policy Optimization for **P**artially-Obs**e**rved **R**eward (HyPeR), a method which estimates the PG using both types of rewards. We show that our PG estimator can substantially reduce the estimation variance compared to typical estimators such as IPS and DR while maintaining unbiasedness under reasonable conditions. In addition, while our approach generally performs effectively using the predefined weight between the target and secondary rewards within the objective, we demonstrate that further enhancement can be achieved by strategically *tuning* the weight to improve the bias-variance trade-off of the PG estimation, which we can perform based only on observable logged data. Finally, we conduct comprehensive experiments on both synthetic and real-world datasets, where the HyPeR algorithm outperforms a range of existing methods in terms of optimizing both the target reward objective and the combined objective of the target and secondary rewards.

The key contributions of our work can be summarized as follows.

- We formulate the general problem of Off-Policy Learning (OPL) with partially-observed rewards, encompassing many prevalent scenarios such as missing data, delayed rewards, data fusion, and censoring, all of which are specific instances of this general problem.
- We propose a new method to address OPL with partial rewards by leveraging more densely-observed secondary rewards to estimate the policy gradient with reduced variance.
- We consider a combined objective defined by a weighted sum of the target and secondary rewards and introduce the novel concept of a strategic use of an incorrect weight in our method to maximize the merit of using the two types of rewards.
- We empirically demonstrate that our method outperforms a range of existing methods in terms of both the target reward objective and combined reward objective.

2 Off-Policy Learning with Partially-Observed Rewards

2.1 The Conventional Formulation

We first formulate the conventional OPL problem regarding the typical contextual bandit setting (Dudík et al., 2014; Swaminathan & Joachims, 2015a; Farajtabar et al., 2018). In the typical formulation, context $x \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$ is a d_x-dimensional vector that is drawn i.i.d. from an unknown distribution $p(x)$. Given context x, a possibly stochastic policy $\pi(a|x)$ chooses an action a within a finite action space denoted here by $\mathcal{A}$. The (target) reward $r \in [0, r_{max}]$ (e.g., ratings, conversions, retention, survivals) is then sampled from an unknown distribution $p(r|x, a)$.

The existing literature defines the objective function in OPL by the expected target reward (often referred to as the *policy value*) as defined below (Swaminathan & Joachims, 2015b).

$$V(\pi) := \mathbb{E}_{p(x)\pi(a|x)p(r|x,a)}[r] = \mathbb{E}_{p(x)\pi(a|x)}[q(x, a)], \tag{1}$$

where $q(x, a) := \mathbb{E}[r \mid x, a]$ is the expected reward given x and a, which we call the *q-function*. In OPL, the goal is to learn a policy π_θ, parameterized by θ, that would maximize the policy value: $\theta^* \in \arg\max_{\theta\in\Theta} V(\pi_\theta)$. In particular, OPL aims to learn such a policy using only logged data consisting of tuples (x, a, r) generated under the *logging* policy denoted by π_0. More specifically, the logged data we can use to perform OPL can be written as $\mathcal{D} = \{(x_i, a_i, r_i)\}_{i=1}^n \sim p(\mathcal{D})$ where the data distribution is induced by the logging policy, i.e., $p(\mathcal{D}) = \prod_{i=1}^n p(x_i)\pi_0(a_i|x_i)p(r_i|x_i, a_i)$.

2.2 Off-Policy Learning via Policy Gradient

Most existing approaches in OPL are based on PG iterations (Ma et al., 2020; Chen et al., 2021). This method updates the policy parameter θ via iterative gradient ascent: $\theta_{t+1} \leftarrow \theta_t + \nabla_\theta V(\pi_\theta)$ where the policy gradient is represented as $\nabla_\theta V(\pi_\theta) = \mathbb{E}_{p(x)\pi_\theta(a|x)}[q(x, a)\nabla_\theta \log \pi_\theta(a|x)]$. This form of PG can be derived by the log-derivative trick and suggests to update the policy parameter so that the resulting policy can choose actions that have high expected rewards with high probability.

To implement the PG iteration, we first need to estimate it, since it is an unknown vector. To achieve this using only the logged data $\mathcal{D}$ collected under the logging policy π_0, the relevant literature relies on estimators such as IPS and DR, which allow for unbiased estimation of the PG (Dudík et al., 2014; Metelli et al., 2021). These PG estimators are defined as follows.

$$\nabla_\theta \widehat{V}_{\mathrm{IPS}}(\pi_\theta; \mathcal{D}) := \frac{1}{n}\sum_{i=1}^n w(x_i, a_i) r_i g_\theta(x_i, a_i), \tag{2}$$

$$\nabla_\theta \widehat{V}_{\mathrm{DR}}(\pi_\theta; \mathcal{D}) := \frac{1}{n}\sum_{i=1}^n w(x_i, a_i)(r_i - \hat{q}(x_i, a_i)) g_\theta(x_i, a_i) + \mathbb{E}_{\pi_\theta(a|x_i)}[\hat{q}(x_i, a) g_\theta(x_i, a)], \tag{3}$$

where $w(x, a) := \pi_\theta(a \mid x)/\pi_0(a \mid x)$ is the importance weight and $g_\theta(x, a) := \nabla_\theta \log \pi_\theta(a \mid x)$ is the policy score function. $\hat{q}(x, a)$ is an estimator of the q-function, which we can obtain by performing reward regression in the logged data $\mathcal{D}$.

These estimators are unbiased (i.e., $\mathbb{E}_{p(\mathcal{D})}[\nabla_\theta \widehat{V}_{\mathrm{IPS}}(\pi_\theta; \mathcal{D})] = \mathbb{E}_{p(\mathcal{D})}[\nabla_\theta \widehat{V}_{\mathrm{DR}}(\pi_\theta; \mathcal{D})] = \nabla_\theta V(\pi_\theta)$) under *full support*, which requires that the logging policy sufficiently explore the action space.

Condition 1 (Full Support)**.** *The logging policy π_0 is said to have full support if $\pi_0(a|x) > 0$ for all $x \in \mathcal{X}$ and $a \in \mathcal{A}$.*

The importance weighting technique used by IPS and DR enables an unbiased estimation of the PG, thereby resulting in effective OPL even without additional exploration. However, if the problem is far from ideal with smaller data sizes, large action spaces, and noisy rewards, existing OPL methods can easily collapse due to substantial variance in PG estimation (Saito & Joachims, 2022).

2.3 Partially-Observed Rewards

As discussed in the introduction and in Table 1, there are many real-life cases where we can observe the target reward only partially. To precisely formulate such a scenario, we introduce an

additional random variable, $o \in \{0, 1\}$, to represent whether the reward is observed for each data point. If $o_i = 1$, the reward r_i is observed; otherwise, it is unavailable, as indicated by setting $r_i = \mathrm{N/A}$. Considering a scenario where the observation indicator comes from an unknown distribution $0 < p(o|x) < 1$, we can generalize the data generating process as $\mathcal{D} = \{(x_i, a_i, o_i, r_i)\}_{i=1}^n \sim p(\mathcal{D}) = \prod_{i=1}^n p(x_i)\pi_0(a_i|x_i)p(o_i|x_i)p(r_i|x_i, a_i, o_i)$, which encompasses many realistic situations like missing data, delayed rewards, data fusion, multi-stage rewards, and censoring.

To implement OPL with such partially-observed rewards, we can simply apply the PG approach using only the data with observed rewards, which we call **r-IPS** and **r-DR**:

$$\nabla_\theta \widehat{V}_{\text{r-IPS}}(\pi_\theta; \mathcal{D}) := \frac{1}{n}\sum_{i=1}^n \frac{o_i}{p(o_i|x_i)} w(x_i, a_i) r_i g_\theta(x_i, a_i), \tag{4}$$

$$\nabla_\theta \widehat{V}_{\text{r-DR}}(\pi_\theta; \mathcal{D}) := \frac{1}{n}\sum_{i=1}^n \frac{o_i}{p(o_i|x_i)} \left\{ w(x_i, a_i)(r_i - \hat{q}(x_i, a_i)) g_\theta(x_i, a_i) + \mathbb{E}_{\pi_\theta(a|x_i)}[\hat{q}(x_i, a) g_\theta(x_i, a)] \right\}, \tag{5}$$

where we can see that these estimators use only the data with reward observations ($o_i = 1$) to estimate the PG. We can readily show that these estimators are unbiased, but they are often substantially inefficient and produce high variance in PG estimation because they use only a part of the data in $\mathcal{D}$ and naively discards all the information when $o_i = 0$.

3 Off-Policy Learning with Secondary Rewards

To deal with the issue of inefficiency of OPL when the rewards are partial, we propose a new formulation of OPL with secondary rewards. In many real-life scenarios, we not only observe the target reward such as the conversion signals that we are optimizing, but also secondary rewards such as clicks and dwell time (Wan & McAuley, 2018; Jadidinejad et al., 2019; Christakopoulou et al., 2022). By leveraging these secondary rewards, we aim to reduce variance in PG estimation, thereby achieving a more efficient OPL even under the challenging scenarios.

To implement this idea, we extend the typical formulation of OPL by introducing the secondary rewards denoted by $s \in \mathbb{R}^{d_s}$, which can be multi-dimensional. We consider the secondary rewards to be sampled from an unknown conditional distribution of the form: $p(s|x, a)$ after taking action a for context x. The logged dataset that we can use in the setting can be written as follows.

$$\mathcal{D} := \{(x_i, a_i, o_i, s_i, r_i)\}_{i=1}^n \sim p(\mathcal{D}) = \prod_{i=1}^n p(x_i)\pi_0(a_i|x_i)p(o_i|x_i)p(s_i|x_i, a_i)p(r_i|x_i, a_i, s_i), \tag{6}$$

In addition to the introduction of secondary rewards, we consider the following generalized objective function called the *combined policy value*, which is defined as a weighted sum of the expected target and secondary rewards, considering a situation where we care about the optimization of the secondary rewards as well.

$$V_c(\pi; \beta) := (1-\beta)V_r(\pi) + \beta V_s(\pi) = (1-\beta)\mathbb{E}_{p(x)\pi(a|x)}[q(x, a)] + \beta \mathbb{E}_{p(x)\pi(a|x)}\left[\sum_{d=1}^{d_s} f_d(x, a)\right], \tag{7}$$

where $f_d(x, a) := \mathbb{E}[s_d|x, a]$ is the d-th dimension of the expected secondary reward, and $\beta \in [0, 1)$ is a parameter to control the importance between the optimization of the target reward and that of the secondary rewards. When $\beta = 0$, this generalized objective reduces to the typical policy value defined in Eq. (1). With some positive value of β, we can address situations where we aim to optimize not only the target reward but also the secondary rewards to some extent.

Given this extended problem of OPL with secondary rewards and with the combined policy value, we can consider a baseline method of using some aggregation of the secondary rewards as a surrogate for the target reward. This baseline approach of using only the secondary rewards as surrogates,

which we call **s-IPS** and **s-DR**, estimates the PG as follows:

$$\nabla_\theta \hat{V}_{\text{s-IPS}}(\pi_\theta; \mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} w(x_i, a_i) F(s_i) g_\theta(x_i, a_i), \tag{8}$$

$$\nabla_\theta \hat{V}_{\text{s-DR}}(\pi_\theta; \mathcal{D}) \tag{9}$$
$$= \frac{1}{n}\sum_{i=1}^{n} \left\{ w(x_i, a_i)(F(s_i) - F(\hat{f}(x_i, a_i)))g_\theta(x_i, a_i) + \mathbb{E}_{\pi_\theta(a|x_i)}[F(\hat{f}(x_i, a))g_\theta(x_i, a)] \right\},$$

where $\hat{f}(x, a)$ is an estimator of $f(x, a)$, and $F(s)$ is some aggregation of the secondary rewards, such as their weighted average, designed to imitate the target reward. By replacing the target reward with $F(s)$, s-IPS and s-DR can use all the data points irrespective of the indicator o_i, thus reducing variance. However, unless the function $F(s)$ accurately describes the target reward, which is untestable, the estimators produce substantial bias against the true PG regarding the target reward, as we will show in our experiments.

4 Hybrid Policy Optimization for Partially-Observed Reward

This section proposes a new OPL algorithm to maximize the combined policy value in Eq. (7) with only partially-observed target rewards and secondary rewards that do not necessarily align accurately with the target reward. We also introduce the concept of strategically using a different value of the balancing factor γ compared to the true value of β in the objective to improve the finite-sample effectiveness of the algorithm.

The key idea behind our algorithm is the introduction of a new estimator of the PG using secondary rewards to reduce variance while also optimizing the secondary rewards depending on the value of β in the objective.

To derive our method, we first focus on the PG estimation regarding the target policy value $V_r(\pi_\theta)$. Particularly to address the variance issue in existing methods, we propose leveraging secondary rewards through the following estimator for $\nabla_\theta V_r(\pi_\theta)$.

$$\nabla_\theta \hat{V}_r(\pi_\theta; \mathcal{D}) := \frac{1}{n}\sum_{i=1}^{n} \Big\{ \mathbb{E}_{\pi_\theta(a|x_i)}\left[\hat{q}(x_i, a) g_\theta(x_i, a)\right] + w(x_i, a_i)\left(\hat{q}(x_i, a_i, s_i) - \hat{q}(x_i, a_i)\right) g_\theta(x_i, a_i)$$
$$+ \frac{o_i}{p(o_i|x_i)} w(x_i, a_i)\left(r_i - \hat{q}(x_i, a_i, s_i)\right) g_\theta(x_i, a_i) \Big\}, \tag{10}$$

where $\hat{q}(x, a, s)$ is an estimator of the q-function conditional on the secondary rewards (i.e., $q(x, a, s)$), which we can derive, e.g., by solving $\hat{q} = \arg\min_{q'} \sum_{(x,a,s,r)\in\mathcal{D}} (r - q'(x, a, s))^2$ using th logged data $\mathcal{D}$. Note also that we can estimate the conditional reward-observation probability $p(o_i|x_i)$ when it is unknown by regressing o to x observed in the logged data $\mathcal{D}$ by a classifier.

We first show that our PG estimator is unbiased under the same conditions as r-DR.

Theorem 1. *(Unbiasedness) Under Condition 1, Eq. (10) is unbiased against the true PG regarding the target reward, i.e.,*

$$\mathbb{E}[\nabla_\theta \hat{V}_r(\pi_\theta; \mathcal{D})] = \nabla_\theta V_r(\pi_\theta) = \nabla_\theta V_c(\pi_\theta; \beta = 0)$$

See Appendix B.1 for the proof.

In addition to its unbiasedness, our PG estimator fully utilizes the information from the secondary rewards, thus reducing the variance compared to r-DR in most cases. The following demonstrates that the variance of Eq. (10) can be much lower than r-DR.

Theorem 2. *(Variance Reduction) Under Condition 1, we have*

$$n(\mathbb{V}_{\mathcal{D}}[\nabla_\theta \hat{V}_{\text{r-DR}}(\pi;\mathcal{D})] - \mathbb{V}_{\mathcal{D}}[\nabla_\theta \hat{V}_r(\pi;\mathcal{D})])$$
$$= \mathbb{E}_{p(x)\pi_0(a|x)p(s|x,a)}\left[\frac{\rho^2}{p(o|x)^2} w(x,a)^2 g_\theta(x,a)^2 \left(\Delta_{q,\hat{q}^{\neg s}}(x,a,s)^2 - \Delta_{q,\hat{q}}(x,a,s)^2\right)\right]$$

where $\rho^2 := \mathbb{V}[o|x]$, $\Delta_{q,\hat{q}^{\neg s}}(x,a,s) := q(x,a,s) - \hat{q}(x,a)$, *and* $\Delta_{q,\hat{q}}(x,a,s) := q(x,a,s) - \hat{q}(x,a,s)$. *See Appendix B.2 for the proof.*

Theorem 2 indicates that there is a reduction in variance if $\hat{q}(x,a,s)$ is better than $\hat{q}(x,a)$ in estimating $q(x,a,s)$, which is often true since secondary rewards are typically somewhat correlated with the target reward. Thus, Eq. (10) is expected to perform better than r-DR while being unbiased under the same condition. To demonstrate this, in the experimental sections, we investigate the difference in performance between Eq. (10), which we name **HyPeR**($\gamma = 0$), and r-DR.

Based on Eq. (10), we introduce our **HyPeR** estimator:

$$\nabla_\theta \hat{V}_{\text{HyPeR}}(\pi_\theta;\mathcal{D},\gamma) = (1-\gamma)\cdot \nabla_\theta \hat{V}_r(\pi_\theta;\mathcal{D}) + \gamma \cdot \nabla_\theta \hat{V}_s(\pi_\theta;\mathcal{D}), \tag{11}$$

where the first term $\nabla_\theta \hat{V}_r(\pi_\theta;\mathcal{D})$ estimates the PG regarding the target policy value $V_r(\pi_\theta)$ as in Eq. (10). The second term $\nabla_\theta \hat{V}_s(\pi_\theta;\mathcal{D})$ estimates the PG regarding the secondary policy value $V_s(\pi_\theta)$, which we can define by applying DR to the sum of secondary rewards as follows:

$$\nabla_\theta \hat{V}_s(\pi_\theta;\mathcal{D})$$
$$= \frac{1}{n}\sum_{i=1}^{n}\left\{\mathbb{E}_{\pi_\theta(a|x_i)}\left[\sum_{d=1}^{d_s}\hat{f}_d(x_i,a)g_\theta(x_i,a_i)\right] + w(x_i,a_i)\sum_{d=1}^{d_s}\left(s_i^d - \hat{f}_d(x_i,a_i)\right)g_\theta(x_i,a_i)\right\}.$$

In Eq. (11), $\gamma \in [0,1)$ is a tunable parameter that defines the mixture ratio between the estimators of the PG regarding the target reward and secondary rewards. When using the predefined weight (i.e., $\gamma = \beta$), Eq. (11) is unbiased against the combined policy gradient (i.e., $\mathbb{E}[\nabla_\theta \hat{V}_{\text{HyPeR}}(\pi_\theta;\mathcal{D})] = \nabla_\theta V(\pi_\theta)$), but the following discusses a strategic tuning of γ to further improve our method.

4.1 Strategic Tuning of the weight γ

Although the natural choice of γ in our PG estimator is β, which defines the true objective in Eq. (7), we argue that *intentionally using $\gamma \neq \beta$ further improves the combined policy value in the finite-sample setting*. This is due to the potential differences in variance between the different rewards. Specifically, strategically shifting weights from the predefined balance, β, can lead to better estimation due to variance reduction, even though it introduces some bias. For example, consider an objective function that is a sum of two estimands, $X + Y$. If estimator $\hat{X}$ carries less variance than estimator $\hat{Y}$, it is likely better to give more weight to $\hat{X}$ to achieve less variance, even at the cost of introducing some bias. When dealing with multiple reward types, this can occur in various situations, such as when one reward is less noisy than the other, or as in our study, when one reward type (secondary reward) is more frequently observed than the other (target reward).

In HyPeR, the PG regarding the target reward $\nabla_\theta \hat{V}_r(\pi_\theta;\mathcal{D})$ often carries much higher variance compared to that regarding the secondary rewards $\nabla_\theta \hat{V}_s(\pi_\theta;\mathcal{D})$ due to the partial-observation nature. Thus, although setting $\gamma = \beta$ will make the HyPeR estimation unbiased and at least reduces the variance from existing methods, it can still have high variance, particularly when β takes a small value. On the other hand, increasing the weight of γ will likely lead to less variance since this prioritizes the second term more, while introducing some bias. This creates a bias-variance trade-off; we aim to tune γ to achieve the optimal combined policy value by the resulting policy:

$$\gamma^* = \underset{\gamma\in[0,1)}{\arg\max}\, V(\pi_\theta(\cdot;\gamma,\mathcal{D});\beta), \tag{12}$$

where $\pi_\theta(\cdot;\gamma,\mathcal{D})$ is a policy optimized under weight γ, and its value is evaluated under the originally defined weight β.

Since the true policy value is unknown, we need to perform Eq. (12) from only the logged dataset $\mathcal{D}$. The most straightforward method is via splitting the dataset $\mathcal{D}$ into training ($\mathcal{D}_{tr}$) and validation ($\mathcal{D}_{val}$) sets. Then, we train a policy using $\mathcal{D}_{tr}$ (i.e., $\pi_\theta(\cdot;\gamma,\mathcal{D}_{tr})$) and estimate the value using $\mathcal{D}_{val}$ (i.e., $\hat{V}(\pi_\theta;\beta,\mathcal{D}_{val})$). However, an issue with this naive procedure is that the policy $\pi_\theta(\cdot;\gamma,\mathcal{D}_{tr})$ is trained on a smaller dataset $\mathcal{D}_{tr}$ instead of the full dataset $\mathcal{D}$ that is used in real training. In the PG estimation, variance is inversely proportional to the data size $|\mathcal{D}|$, while bias is unaffected. Therefore, the naive tuning procedure is expected to end up selecting a higher weight than the optimal weight γ^*. To address this issue, we ensure that $|\mathcal{D}_{tr}| = |\mathcal{D}|$ through bootstrapping:

$$\mathcal{D}'_{tr} = \big\{(x'_i, a'_i, s'_i, o'_i, r'_i) \mid i = 1, ..., n\big\}, \tag{13}$$

where $\{(x'_i, a'_i, s'_i, o'_i, r'_i)\}_{i=1}^n \sim \mathcal{D}_{tr}$ is sampled independently *with replacement*. By doing so, we alleviate the issue regarding the variance estimation, since $\mathcal{D}'_{tr}$ now has the same size as the full dataset $\mathcal{D}$. Using $\mathcal{D}'_{tr}$, we propose to solve the bi-level optimization to data-drivenly tune γ:

$$\hat{\gamma}^* = \arg\max_{\gamma\in[0,1]} \hat{V}(\pi_\theta(\cdot;\gamma,\mathcal{D}'_{tr});\beta,\mathcal{D}_{val}). \tag{14}$$

Note that when performing hyperparameter tuning in OPL, it is known that overestimation of the policy value in the validation set can lead to the failure of a typical or naive tuning procedure, such as using standard estimators like IPS and DR (Saito & Nomura, 2024). Although our proposed procedure performs nearly optimally in our experiments, it may benefit from incorporating correction methods to achieve more effective parameter tuning for γ in practice.

5 Synthetic Experiment

This section evaluates HyPeR on synthetic data to identify the situations where HyPeR's performance becomes particularly promising.

5.1 Data Setup

To create the synthetic data, we sample 10-dimensional context vectors x from a standard normal distribution, and the sample size is fixed at $n = 2000$ by default. We synthesize the logging policy as $\pi_0 = \mathrm{softmax}(\phi(x^T\mathcal{M}_{X,A}a + x^T\theta_x + a^T\theta_a))$, where $\mathcal{M}, \theta_x, \theta_a$ are parameter matrices randomly sampled from a uniform distribution with range $[-1, 1]$, and actions $a \in \mathcal{A}$ are sampled following this logging policy ($|\mathcal{A}| = 10$). ϕ is a parameter that controls how deterministic the logging policy would become, and we set this at $\phi = -2.0$ by default. Then we synthesize each dimension of the 5-dimensional expected secondary rewards given x and a as

$$f_d(x, a) = x^T\mathcal{M}'_{X,A}a + x^T\theta'_x + a^T\theta'_a, \tag{15}$$

and secondary rewards s are sampled from a normal distribution $s_d \sim \mathcal{N}(f_d(x, a), \sigma_s^2)$. In the main text, we set the default to $\sigma_s = 0.5$, and the results for other values of σ_s can be found in Appendix C.2. Then, we synthesize the expected target reward function as

$$q(x, a, f(x, a)) := (1-\lambda)(x^T\mathcal{M}''_{X,A}a + x^T\theta''_x + a^T\theta''_a + x^T\mathcal{M}_{X,F}f + a^T\mathcal{M}_{A,F}f) + \lambda f^T\theta_f. \tag{16}$$

$\lambda \in [0, 1]$ is an experimental parameter to control how much secondary rewards are correlated to the target reward. When $\lambda = 1$, secondary rewards are completely correlated with the target reward, while a smaller λ will make it less correlated. We use $\lambda = 0.7$ as a default setting throughout the synthetic experiment. The target reward is then sampled from a normal distribution as $r \sim \mathcal{N}(q(x, a, f(x, a)), \sigma_r^2)$ with default $\sigma_r = 0.5$ (results with other σ_r values are provided in Appendix C.2). The target reward observation probability is an experimental parameter and is set to $p(o|x) = 0.2$ for all x by default. The true weight β, which is used to define the combined policy value $V_c(\pi;\beta)$, is set to $\beta = 0.3$ and is also one of the experimental parameters.

In the synthetic experiments, we generally use the predefined weight β for HyPeR, and write it as HyPeR($\gamma = \beta$). We compare the HyPeR algorithm against HyPeR($\gamma = 0$), r-IPS, r-DR, s-IPS and s-DR. For s-IPS and s-DR, we use $F(s) = s^T(\theta_f + \varepsilon_F)$, where θ_f is identical to that of Eq. (16), and $\varepsilon_F \sim \mathcal{N}(0, \sigma_F^2)$ is the noise added to simulate the inaccuracy of describing the target reward (set to $\sigma_F = 0.4$).

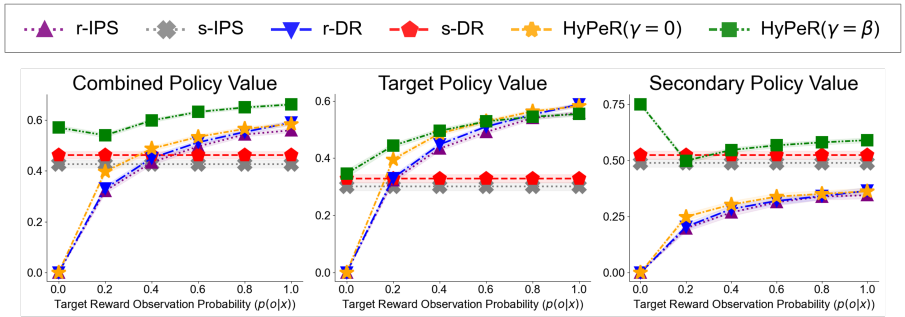

Figure 1: Comparing the combined, target, and secondary policy values of OPL methods with **varying target reward observation probabilities** $(p(o|x))$.

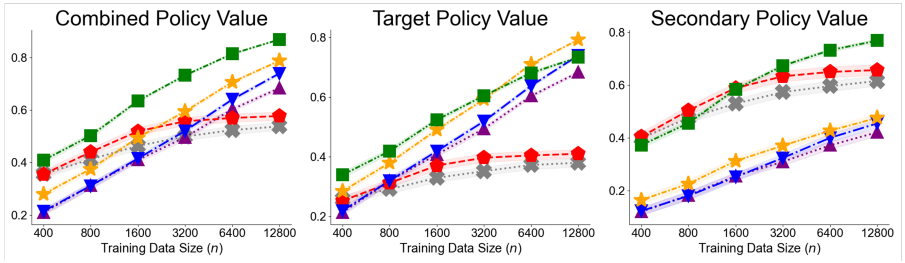

Figure 2: Comparing the combined, target, and secondary policy values of OPL methods with **varying training data sizes** (n).

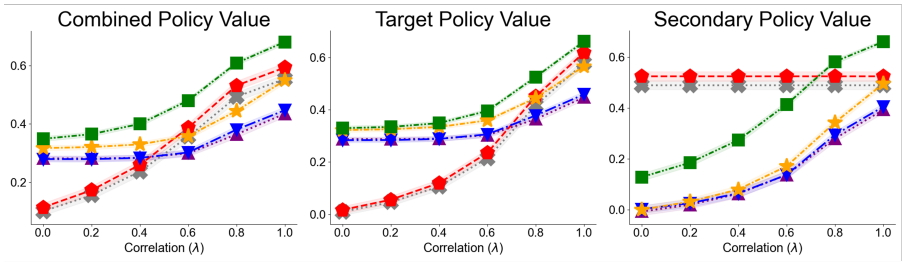

Figure 3: Comparing the combined, target, and secondary policy values of OPL methods with **varying degrees of target-secondary reward correlation** (λ). Secondary rewards can completely explain the target reward at $\lambda = 1$, and they are less correlated with smaller λ.

5.2 Results and Discussion

We run OPL simulations 100 times with different train-test splits. We report the relative policy values calculated by $(V(\pi^*) - V(\pi_\theta))/(V(\pi_\theta) - V(\pi_{\text{unif}}))$, where π^* is the optimal policy and π_{unif} is a uniform random policy. This way, the performance of a random policy will have the value of 0 and an optimal policy would have the value 1, thus easier to interpret. Note that the shaded regions in the plots represent 95% confidence intervals estimated via bootstrapping.

How does HyPeR perform with varying target reward observation probabilities? Figure 1 evaluates the relative policy value when we vary the observation probability of the target reward $p(o|x)$. We observe that HyPeR($\gamma = \beta$) provides the highest combined policy value in all cases. It is also able to optimize both the target and secondary policy values in a balanced way. In addition, HyPeR($\gamma = \beta$) outperforms all the baselines regarding the target policy value, and it even outperforms HyPeR($\gamma = 0$) when the observation probability is low. This is because the secondary reward maximization component of HyPeR reduces variance of the estimation, as secondary rewards are denser. We also observe that the HyPeR($\gamma = 0$) (target reward maximization component of HyPeR) is always performing better than r-DR, which suggests the use of secondary rewards enhances the estimation of the target gradient $\nabla_\theta V_r(\pi_\theta)$ alone, as in Theorem 2.

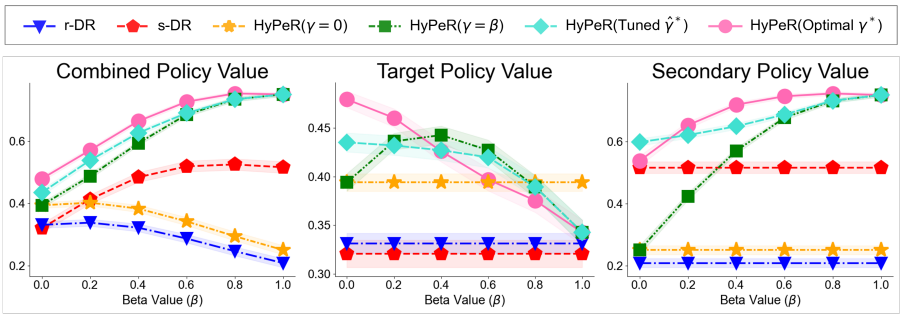

Figure 4: Comparing the combined, target, and secondary policy values of OPL methods with **varying** β, which is a weight that balances target and secondary policy values in the combined policy value. Higher β means the secondary policy value becomes more dominant.

How does HyPeR perform with varying training data sizes? Figure 2 evaluates the methods' policy values with different training data sizes. Larger data size generally makes gradient estimations more accurate as it decreases the variance. Therefore, as the data size increases, the methods that use the target reward in their estimation (i.e., HyPeR($\gamma = \beta$), HyPeR($\gamma = 0$), r-IPS, and r-DR) perform increasingly better compared to the ones that do not (i.e., s-IPS and s-DR). The left plot in Figure 2 shows that HyPeR($\gamma = \beta$) performs the best in most cases regarding the combined policy value $V_c(\pi)$. It even performs the best in terms of the target policy value $V_r(\pi)$ in most cases, particularly when the data size is small due to the use of secondary rewards and variance reduction.

How does HyPeR perform with varying correlation between target and secondary rewards? Figure 3 shows the results with different degrees of correlation between the secondary rewards and the target reward, controlled by λ in Eq. (16). The larger λ indicates that the target reward can be more predictable by the secondary reward. We can see from the figure that HyPeR($\gamma = \beta$) is the best option in almost all cases for both the combined policy value and target policy value. We also observe that HyPeR($\gamma = \beta$), s-IPS, and s-DR perform comparatively well when secondary rewards are more correlated (larger λ), as they find more advantage of using the secondary rewards as surrogates.

How do HyPeR and data-driven weight selection perform with varying weight β? Section 4.1 showed that a data-driven tuning of the weight γ will potentially improve the performance of HyPeR. To empirically evaluate whether our weight-tuning method can make further improvement, in addition to HyPeR($\gamma = \beta$), we add HyPeR(Tuned $\hat{\gamma}^*$) and HyPeR(Optimal γ^*) for comparison. HyPeR(Tuned $\hat{\gamma}^*$) estimates the optimal weight through estimation using the method described in Section 4.1. HyPeR(Optimal γ^*) is a skyline that performs HyPeR with a truly optimal weight γ^*, which is not feasible in practice but it is a useful reference. Due to the increase in number of comparisons, here we reduce r-IPS and s-IPS from the baselines, which always perform worse than DR-based methods (Appendix C.1 shows the complete results including r-IPS and s-IPS). Figure 4 provides the results, including HyPeR(Tuned $\hat{\gamma}^*$) and HyPeR(Optimal γ^*), with varying weight β. β is a weight which determines the relative importance of the secondary policy value compared to the target policy value in the combined policy value $V_c(\pi)$. From the results, we can see that HyPeR(Tuned $\hat{\gamma}^*$) always outperforms HyPeR($\gamma = \beta$) and all other feasible methods, suggesting the effectiveness of our weight tuning procedure. This observation also interestingly implies that an incorrect weight (i.e., $\hat{\gamma}^* \neq \beta$) can lead to a better policy performance compared to $\gamma = \beta$, due to the variance reduction at the cost of some bias as discussed in Section 4.1.

6 Real-World Experiment

To assess the real-world applicability of HyPeR, we now evaluate it on KuaiRec (Gao et al., 2022), a publicly available fully-observed user-item matrix dataset collected on a short video platform, where 1,411 users have viewed all 3317 videos and left play duration as feedback. This feature of the dataset enables an OPL experiment to be conducted without synthesizing the reward function.

Setup. We use watch ratio (= watch duration/video length) as target reward r. We use four dimensional secondary rewards; each dimension has a realistic reason why the platform would want to maximize it. The first dimension is binary, where $s_1 = 1$ if $r \geq 2.0$, and $s_1 = 0$ if $r < 2.0$; this

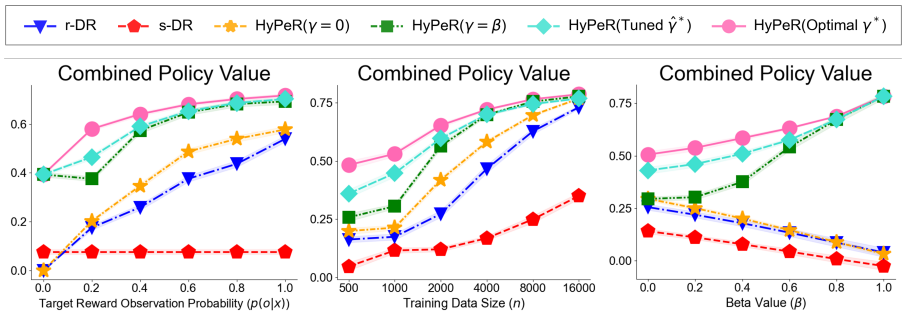

Figure 5: Comparing the **combined policy values** of OPL methods under (**left**) varying target reward observation probabilities ($p(o|x)$), (**center**) varying data sizes (n), and (**right**) varying weights β on the **KuaiRec** dataset.

reward maximization lets the platform prioritize sessions with exceptionally long watch ratio. The second is also binary, where $s_2 = -1$ if $r < 0.5$, and $s_2 = 0$ if $r \geq 0.5$. This dimension is built to strictly punish and refrain from sessions with exceptionally low watch ratio (raising the engagement floor). The third dimension of the secondary rewards is time since video upload (multiplied by -1), which is implemented to prioritize newer videos over older ones. The last dimension is video length, which is designed to prioritize longer videos when watch ratios are similar. Note that all continuous rewards are normalized to the range $[0, 1]$. We use the first dimension of the secondary rewards to express $F(s)$ for s-IPS and s-DR.

To compare the different methods, we randomly choose 988 users (70%) for training and 423 users (30%) for evaluation. We set the target reward observation probability to $p(o|x) = 0.2$ for all x, training data size to $n = 1000$, and weight $\beta = 0.3$, as default experimental parameters. The actions are chosen randomly with size $|\mathcal{A}| = 100$, and Appendix C.3 shows results with varying numbers of actions. We run 100 simulations with different train-test splits and different action spaces. The logging policy was synthesized as $\pi_0 = \mathrm{softmax}(\phi(x^T \mathcal{M}_{X,A} a + x^T \theta_x + a^T \theta_a))$, with $\phi = -2.0$.

Results. Figure 5 provides real-world experiment results with varying target reward observation probabilities $p(o|x)$, varying data sizes n, and varying weights β in the combined policy value. In this section, we provide comparisons of the combined policy values for r-DR, s-DR, HyPeR($\gamma = 0$), HyPeR($\gamma = \beta$), HyPeR(Tuned $\hat{\gamma}^*$) and HyPeR(Optimal γ^*). In Figure 5, we observe that HyPeR($\gamma = \beta$) and HyPeR(Tuned $\hat{\gamma}^*$) both outperform the baselines by far. Moreover, by comparing HyPeR(Tuned $\hat{\gamma}^*$) against HyPeR($\gamma = \beta$), we could see that intentionally using the tuned weight leads to a better performance, particularly under more challenging scenarios (i.e., low primary reward observation and small training data size). We also observe, in Appendix C.3, that HyPeR outperforms the baseline methods in terms of not only the combined policy value, but also the target and secondary policy values in the real-world experiment. It is also interesting that, in terms of the target policy value, the policy performance becomes much worse when we use only the secondary rewards (i.e., HyPeR's performance at $\beta = 1$), compared to the case where we use only the target reward (i.e., performance of r-DR). This demonstrates that the sum of the secondary rewards that we used in our real experiment is only slightly correlated with the target reward.

7 Conclusion and Future Work

This paper studies the general problem of off-policy learning (OPL) in contextual bandits with partially-observed target rewards. Relying only on the target reward to learn a policy often leads to a high variance and inefficient learning in this setup. We propose **Hybrid Policy Optimization for Partially-Observed Reward** (**HyPeR**), a novel approach that integrates secondary rewards to improve policy gradient estimation. HyPeR effectively reduces variance while maintaining unbiasedness and can be easily extended to deal with the combined objective of maximizing both the target and secondary rewards from available data. Experiments on synthetic and real-world datasets demonstrate that HyPeR outperforms existing methods in optimizing both the target and combined policy values. As future work, it would be valuable to extend our formulation and method to the problem of offline reinforcement learning. Conducting an experiment in a real environment would also further demonstrate the effectiveness, applicability, and real-world impact of our method.

REFERENCES

Shima Alizadeh, Aniruddha Bhargava, Karthick Gopalswamy, Lalit Jain, Branislav Kveton, and Ge Liu. Pessimistic off-policy multi-objective optimization. October 2023. URL http://arxiv.org/abs/2310.18617. arXiv preprint arXiv:2310.18617.

Javier Arroyo, Francesco Corea, Guillermo Jimenez-Diaz, and Juan A. Recio-Garcia. Assessment of machine learning performance for decision support in venture capital investments. *IEEE Access*, 7:124233–124243, 2019. doi: 10.1109/ACCESS.2019.2938659.

Azin Ashkan and Charles L.A. Clarke. Characterizing commercial intent. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pp. 67–76, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585123. doi: 10.1145/1645953.1645965. URL https://doi.org/10.1145/1645953.1645965.

W. S. Barnett. Effectiveness of early educational intervention. *Science*, 333(6045):975–978, 2011. doi: 10.1126/science.1204534. URL https://www.science.org/doi/abs/10.1126/science.1204534.

Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed Chi. Top-k off-policy correction for a reinforce recommender system, 2021.

Konstantina Christakopoulou, Can Xu, Sai Zhang, Sriraj Badam, Trevor Potter, Daniel Li, Hao Wan, Xinyang Yi, Ya Le, Chris Berg, Eric Bencomo Dixon, Ed H. Chi, and Minmin Chen. Reward shaping for user satisfaction in a reinforce recommender, 2022.

Miroslav Dudík, Dumitru Erhan, John Langford, and Lihong Li. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4), November 2014. ISSN 0883-4237. doi: 10.1214/14-sts500. URL http://dx.doi.org/10.1214/14-STS500.

Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. More robust doubly robust off-policy evaluation, 2018. URL https://arxiv.org/abs/1802.03493.

Germano Gabbianelli, Gergely Neu, and Matteo Papini. Importance-weighted offline learning done right. In *International Conference on Algorithmic Learning Theory*, pp. 614–634. PMLR, 2024.

Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. Kuairec: A fully-observed dataset and insights for evaluating recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, pp. 540–550, 2022. doi: 10.1145/3511808.3557220. URL https://doi.org/10.1145/3511808.3557220.

Guy Hadash, Oren Sar Shalom, and Rita Osadchy. Rank and rate: multi-task learning for recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18. ACM, September 2018. doi: 10.1145/3240323.3240406. URL http://dx.doi.org/10.1145/3240323.3240406.

Guido Imbens, Nathan Kallus, Xiaojie Mao, and Yuhao Wang. Long-term causal inference under persistent confounding via data combination. *arXiv preprint arXiv:2202.07234*, 2022.

Amir H. Jadidinejad, Craig Macdonald, and Iadh Ounis. Unifying explicit and implicit feedback for rating prediction and ranking recommendation tasks. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '19, pp. 149–156, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450368810. doi: 10.1145/3341981.3344225. URL https://doi.org/10.1145/3341981.3344225.

Janus Christian Jakobsen, Christian Gluud, Jørn Wetterslev, and Per Winkel. When and how should multiple imputation be used for handling missing data in randomised clinical trials – a practical guide with flowcharts. *BMC Medical Research Methodology*, 17(1):162, December 2017. ISSN 1471-2288. doi: 10.1186/s12874-017-0442-1.

Olivier Jeunen and Bart Goethals. Pessimistic reward models for off-policy learning in recommendation. In *Proceedings of the 15th ACM Conference on Recommender Systems*, RecSys '21, pp. 63–74, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384582. doi: 10.1145/3460231.3474247. URL https://doi.org/10.1145/3460231.3474247.

Thorsten Joachims, Adith Swaminathan, and Maarten de Rijke. Deep learning with logged bandit feedback. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJaP_-xAb.

Sofia Ira Ktena, Alykhan Tejani, Lucas Theis, Pranay Kumar Myana, Deepak Dilipkumar, Ferenc Huszar, Steven Yoo, and Wenzhe Shi. Addressing delayed feedback for continuous training with neural networks in ctr prediction, 2021. URL https://arxiv.org/abs/1907.06558.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

Nathan N. Liu, Evan W. Xiang, Min Zhao, and Qiang Yang. Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pp. 1445–1448, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450300995. doi: 10.1145/1871437.1871643. URL https://doi.org/10.1145/1871437.1871643.

Ben London and Ted Sandler. Bayesian counterfactual risk minimization. In *International Conference on Machine Learning*, pp. 4125–4133. PMLR, 2019.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Ji Yang, Minmin Chen, Jiaxi Tang, Lichan Hong, and Ed H. Chi. Off-policy learning in two-stage recommender systems. In *Proceedings of The Web Conference 2020*, WWW '20, pp. 463–473, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370233. doi: 10.1145/3366423.3380130. URL https://doi.org/10.1145/3366423.3380130.

Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 1137–1140, 2018.

Alberto Maria Metelli, Alessio Russo, and Marcello Restelli. Subgaussian and differentiable importance sampling for off-policy evaluation and learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8119–8132. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/4476b929e30dd0c4e8bdbcc82c6ba23a-Paper.pdf.

Abhishek Pratap, Elias Chaibub Neto, Phil Snyder, Carl Stepnowsky, Noémie Elhadad, Daniel Grant, Matthew H. Mohebbi, Sean Mooney, Christine Suver, John Wilbanks, Lara Mangravite, Patrick J. Heagerty, Pat Areán, and Larsson Omberg. Indicators of retention in remote digital health studies: a cross-study evaluation of 100,000 participants. *npj Digital Medicine*, 3(1):21, 2 2020. ISSN 2398-6352. doi: 10.1038/s41746-020-0224-8. URL https://doi.org/10.1038/s41746-020-0224-8.

Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, Lin Qiu, and Yong Yu. Deep recurrent survival analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33 (01):4798–4805, Jul. 2019. doi: 10.1609/aaai.v33i01.33014798. URL https://ojs.aaai.org/index.php/AAAI/article/view/4407.

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

Noveen Sachdeva, Lequn Wang, Dawen Liang, Nathan Kallus, and Julian McAuley. Off-policy evaluation for large action spaces via policy convolution, 2023. URL https://arxiv.org/abs/2310.15433.

Yuta Saito. Doubly robust estimator for ranking metrics with post-click conversions. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pp. 92–100, 2020.

Yuta Saito and Thorsten Joachims. Counterfactual learning and evaluation for recommender systems: Foundations, implementations, and recent advances. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pp. 828–830, 2021.

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

Yuta Saito and Masahiro Nomura. Hyperparameter optimization can even be harmful in off-policy learning and how to deal with it. *arXiv preprint arXiv:2404.15084*, 2024.

Yuta Saito, Himan Abdollahpouri, Jesse Anderton, Ben Carterette, and Mounia Lalmas. Long-term off-policy evaluation and learning. In *Proceedings of the ACM on Web Conference 2024*, pp. 3432–3443, 2024.

Otmane Sakhi, Imad Aouali, Pierre Alquier, and Nicolas Chopin. Logarithmic smoothing for pessimistic off-policy evaluation, selection and learning. *arXiv preprint arXiv:2405.14335*, 2024.

Yi Su, Lequn Wang, Michele Santacatterina, and Thorsten Joachims. Cab: Continuous adaptive blending estimator for policy evaluation and learning, 2019. URL https://arxiv.org/abs/1811.02672.

Yi Su, Maria Dimakopoulou, Akshay Krishnamurthy, and Miroslav Dudík. Doubly robust off-policy evaluation with shrinkage, 2020a.

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

Adith Swaminathan and Thorsten Joachims. The self-normalized estimator for counterfactual learning. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pp. 3231–3239, Cambridge, MA, USA, 2015a. MIT Press.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback, 2015b. URL https://arxiv.org/abs/1502.02362.

Muhammad Faaiz Taufiq, Arnaud Doucet, Rob Cornish, and Jean-Francois Ton. Marginal density ratio for off-policy evaluation in contextual bandits. *Advances in Neural Information Processing Systems*, 36, 2024.

Takuma Udagawa, Haruka Kiyohara, Yusuke Narita, Yuta Saito, and Kei Tateno. Policy-adaptive estimator selection for off-policy evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10025–10033, 2023.

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

Mengting Wan and Julian McAuley. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, pp. 86–94, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450359016. doi: 10.1145/3240323.3240369. URL https://doi.org/10.1145/3240323.3240369.

Nan Wang, Hongning Wang, Maryam Karimzadehgan, Branislav Kveton, and Craig Boutilier. Imo^3: Interactive multi-objective off-policy optimization. January 2022a. URL http://arxiv.org/abs/2201.09798. arXiv preprint arXiv:2201.09798.

Ping Wang, Yan Li, and Chandan K. Reddy. Machine learning for survival analysis: A survey. *ACM Comput. Surv.*, 51(6), feb 2019a. ISSN 0360-0300. doi: 10.1145/3214306. URL https://doi.org/10.1145/3214306.

Siyu Wang, Xiaocong Chen, Dietmar Jannach, and Lina Yao. Causal decision transformer for recommender systems via offline reinforcement learning, 2023. URL https://arxiv.org/abs/2304.07920.

Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. Doubly robust joint learning for recommendation on data missing not at random. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6638–6647. PMLR, 09–15 Jun 2019b. URL https://proceedings.mlr.press/v97/wang19n.html.

Yuyan Wang, Mohit Sharma, Can Xu, Sriraj Badam, Qian Sun, Lee Richardson, Lisa Chung, Ed H. Chi, and Minmin Chen. Surrogate for long-term user experience in recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, pp. 4100–4109, New York, NY, USA, 2022b. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539073. URL https://doi.org/10.1145/3534678.3539073.

A Related Work

A.1 Off-Policy Learning

The contextual bandit framework has been used as one of the favored methods for online decision-making under uncertainty (Lattimore & Szepesvári, 2020). There is also a growing demand to refine decision-making only using historical datasets without needing to perform active exploration. Consequently, off-policy learning (OPL) methods, which aim to learn a decision-making policy using only historical datasets collected under a different policy in the contextual bandit framework, has attracted attention (Joachims et al., 2018). This area of research could be applied to many real-world interactive systems like recommender systems and ad placement.

There are mainly two families of approaches in OPL: regression-based approach and policy-based approach. The regression-based method uses a regression estimate that was trained on the data to predict the rewards from the logged data (Sachdeva et al., 2020; Jeunen & Goethals, 2021). The policy is then made by simply choosing the action with the highest predicted reward deterministically or from a distribution based on the estimates. A drawback of this approach is the bias formed by the inaccurate regression estimation. In contrast, the policy-based approach aims to update the policy by gradient ascent iterations. The policy gradient needs to be estimated from data using techniques like IPS and DR (Dudík et al., 2014; Su et al., 2020a). These estimations are unbiased under some assumptions, but on platforms where rewards are only partially observed, these estimators often suffer from very high variance (Saito & Joachims, 2021; Sachdeva et al., 2023; Taufiq et al., 2024). One possible approach to address this variance issue is through the use of secondary rewards that are more frequently observed instead, which lowers variance by large. However, as secondary rewards often misalign with the target rewards, this suffers greatly from bias. Targeting this problem, HyPeR effectively uses both types of rewards to enhance the estimation of the gradient.

We did not compare pessimistic OPL methods in our experiments because they are not relevant to our context (Swaminathan & Joachims, 2015a; London & Sandler, 2019; Gabbianelli et al., 2024; Sakhi et al., 2024). Our main motivation is to address the prevalent problem of partial target rewards, an issue that pessimistic techniques are not designed to solve. However, our proposed method could easily be combined with a pessimistic approach if desired. It is also important to note that when tuning hyperparameters in OPL, there is the issue of overestimation in the validation estimate of the policy value. In such cases, it may be beneficial to use correction methods as proposed in (Saito & Nomura, 2024). Additionally, the line of work on adaptive estimator selection (Su et al., 2020b; Udagawa et al., 2023) is relevant in this context. However, their primary focus is on data-adaptive estimator selection for OPE, rather than on effectively tuning hyperparameters for OPL.

Our work is also related to the multi-objective optimization problems (Wang et al., 2022a; Alizadeh et al., 2023). However, these works solely focus on achieving weights that meet designer's desired balance of multiple rewards, where the rewards generally have no relation to each other. In contrast, our work is crucially different from the fact that we use secondary rewards to *enhance* the estimation of the target rewards and the combined reward function, particularly in a practical situation where target rewards are only partially observed.

A.2 Partial Target Rewards and Secondary Rewards

Our work is related to various fields that deal with partial observations of rewards due to reasons like missing data (Jakobsen et al., 2017; Wang et al., 2019b; Jadidinejad et al., 2019; Christakopoulou et al., 2022), censoring (Ren et al., 2019; Wang et al., 2019a), delayed observation (Wang et al., 2022b; 2023; Saito et al., 2024), data fusion (Imbens et al., 2022), and multi-stage rewards (Wan & McAuley, 2018; Hadash et al., 2018). Problems of partially-observed data have been addressed in various studies. For example, Liu et al. (2010) propose to unify implicit and explicit feedback in a recommendation setting, to enhance collaborative filtering. Wan & McAuley (2018) have addressed the implicit-explicit feedback problem by modeling them into a monotonic behavior chain and considering them as multi-stage rewards. Christakopoulou et al. (2022) have proposed to design a reinforcement learning model that uses user trajectories (secondary reward) to maximize user satisfaction (target reward). Wang et al. (2022b) investigate the relationship between short-term surrogates and long-term user satisfaction.

In the field of off-policy evaluation and learning, perhaps the closest setting to our work is LOPE (Saito et al., 2024), which aims to estimate the long-term performance of the evaluation policy. To achieve this, in addition to the long-term rewards from historical datasets, Saito et al. (2024) leverage short-term rewards from an *online* experiment. However, while Saito et al. (2024) specifically focus on such setting where short-term online experiments can be done, we do not assume access to such an online experiment data. In addition, under their formulation, the two types of rewards have no difference in observation density, i.e., the long-term reward is NOT partial.

B Proofs and Derivations

B.1 Proof of Theorem 1

Proof. To show the unbiasedness of the estimator Eq. (10), we show that its expectation is equal to the true policy gradient given in Eq. (1).

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{D}}[\nabla_\theta \hat{V}_r(\pi_\theta; D)] \\
&= \mathbb{E}_{\mathcal{D}}\Bigg[\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\pi_\theta(a|x_i)}\left[\hat{q}(x_i,a) g_\theta(x_i,a)\right] \\
&\quad + \frac{\pi_\theta(a_i|x_i)}{\pi_0(a_i|x_i)}\left(\hat{q}(x_i,a_i,s_i) - \hat{q}(x_i,a_i)\right) g_\theta(x_i,a_i) \\
&\quad + \frac{o_i}{p(o|x)}\frac{\pi_\theta(a_i|x_i)}{\pi_0(a_i|x_i)}\left(r_i - \hat{q}(x_i,a_i,s_i)\right) g_\theta(x_i,a_i)\Bigg] \\
&= \mathbb{E}_{p(x)}\Bigg[\mathbb{E}_{\pi_\theta(a|x)}\left[\hat{q}(x,a) g_\theta(x,a)\right] \\
&+ \mathbb{E}_{\pi_0(a|x)p(s|x,a)}\left[\frac{\pi_\theta(a|x)}{\pi_0(a|x)}\left(\hat{q}(x,a,s) - \hat{q}(x,a)\right) g_\theta(x,a)\right] \\
&+ \mathbb{E}_{\pi_0(a|x)p(s|x,a)p(o|x)p(r|x,a,s)}\left[\frac{o}{p(o|x)}\frac{\pi_\theta(a|x)}{\pi_0(a|x)}\left(r - \hat{q}(x,a,s)\right) g_\theta(x,a)\right]\Bigg] \\
&= \mathbb{E}_{p(x)}\Bigg[\mathbb{E}_{\pi_\theta(a|x)}\left[\hat{q}(x,a) g_\theta(x,a)\right] \\
&+ \mathbb{E}_{\pi_0(a|x)}\left[\frac{\pi_\theta(a|x)}{\pi_0(a|x)}\left(\hat{q}(x,a,f(x,a)) - \hat{q}(x,a)\right) g_\theta(x,a)\right] \\
&+ \mathbb{E}_{\pi_0(a|x)p(o|x)}\left[\frac{o}{p(o|x)}\frac{\pi_\theta(a|x)}{\pi_0(a|x)}\left(q(x,a,f(x,a)) - \hat{q}(x,a,f(x,a))\right) g_\theta(x,a)\right]\Bigg] \\
&= \mathbb{E}_{p(x)}\Bigg[\mathbb{E}_{\pi_\theta(a|x)}\left[\hat{q}(x,a) g_\theta(x,a)\right] \\
&+ \mathbb{E}_{\pi_\theta(a|x)}\left[\left(\hat{q}(x,a,f(x,a)) - \hat{q}(x,a)\right) g_\theta(x,a)\right] \\
&+ \mathbb{E}_{\pi_0(a|x)}\left[\frac{\pi_\theta(a|x)}{\pi_0(a|x)}\left(q(x,a,f(x,a)) - \hat{q}(x,a,f(x,a))\right) g_\theta(x,a)\right]\Bigg] \\
&= \mathbb{E}_{p(x)\pi_\theta(a|x)}\left[q(x,a,f(x,a))\right] = \nabla_\theta V_r(\pi)
\end{aligned}
$$

□

B.2 Proof of Theorem 2

Proof. Using the law of total variance, the difference in variance of the j-th element of the gradient estimator can be decomposed into

$$
\begin{aligned}
& n(\mathbb{V}_{\mathcal{D}}[\nabla_\theta \hat{V}_{\text{r-DR}}(\pi_\theta; D)] - \mathbb{V}_{\mathcal{D}}[\nabla_\theta \hat{V}_{\text{r}}(\pi_\theta; D)]]) \\
&= \mathbb{E}_{p(x)\pi_0(a|x)p(s|x,a)p(o|x)}\left[\left(\frac{o}{p(o|x)} w(x,a) g^{(j)}{}_\theta(x,a)\right)^2 \sigma^2(x,a,s)\right] \\
&\quad + \mathbb{V}_{p(x)\pi_0(a|x)p(s|x,a)p(o|x)}\Bigg[\mathbb{E}_{p(r|x,a,s)}\Big[\mathbb{E}_{\pi_\theta(a|x)}[\hat{q}(x,a) g^{(j)}{}_\theta(x,a)] \\
&\qquad + \frac{o}{p(o|x)} w(x,a)(r - \hat{q}(x,a)) g^{(j)}{}_\theta(x,a)\Big]\Bigg] \\
&\quad - \mathbb{E}_{p(x)\pi_0(a|x)p(s|x,a)p(o|x)}\left[\left(\frac{o}{p(o|x)} w(x,a) g^{(j)}{}_\theta(x,a)\right)^2 \sigma^2(x,a,s)\right] \\
&\quad - \mathbb{V}_{p(x)\pi_0(a|x)p(s|x,a)p(o|x)}\Bigg[\mathbb{E}_{p(r|x,a,s)}\Big[\mathbb{E}_{\pi_\theta(a|x)}[\hat{q}(x,a) g^{(j)}{}_\theta(x,a)] \\
&\qquad + w(x,a)(\hat{q}(x,a,s) - \hat{q}(x,a)) g^{(j)}{}_\theta(x,a) \\
&\qquad + \frac{o}{p(o|x)} w(x,a)(r - \hat{q}(x,a,s)) g^{(j)}{}_\theta(x,a)\Big]\Bigg],
\end{aligned}
$$

where $\sigma^2(x,a,s) := \mathbb{V}[r|x,a,s]$, and $g_i^{(j)}(x,a)$ is the policy score function of the j-th element. We now focus on each of the terms. After, canceling out the first and the third term, the second term further simplifies using the law of total variance:

$$
\begin{aligned}
& \mathbb{E}_{p(x)\pi_0(a|x)p(s|x,a)}\left[\frac{\rho^2}{p(o|x)^2} w^2(x,a) g^{(j)}{}_\theta(x,a)^2 \Delta_{q,\hat{q}^{\neg s}}(x,a,s)^2\right] \\
&\quad + \mathbb{V}_{p(x)\pi_0(a|x)p(s|x,a)}\Bigg[\mathbb{E}_{\pi_\theta(a|x)}[\hat{q}(x,a) g^{(j)}{}_\theta(x,a)] \\
&\qquad + w(x,a)(q(x,a,s) - \hat{q}(x,a)) g^{(j)}{}_\theta(x,a)\Bigg],
\end{aligned} \tag{17}
$$

where $\rho^2 := \mathbb{V}[o|x]$, and $\Delta_{q,\hat{q}^{\neg s}}(x,a,s) := q(x,a,s) - \hat{q}(x,a)$. The fourth term can simplify into

$$
\begin{aligned}
& -\Bigg(\mathbb{E}_{p(x)\pi_0(a|x)p(s|x,a)}\left[\frac{\rho^2}{p(o|x)^2} w^2(x,a) g^{(j)}{}_\theta(x,a)^2 \Delta_{q,\hat{q}}(x,a,s)^2\right] \\
&\quad + \mathbb{V}_{p(x)\pi_0(a|x)p(s|x,a)}\Bigg[\mathbb{E}_{\pi_\theta(a|x)}[\hat{q}(x,a) g^{(j)}{}_\theta(x,a)] \\
&\qquad + w(x,a)(q(x,a,s) - \hat{q}(x,a)) g^{(j)}{}_\theta(x,a)\Bigg]\Bigg) \; notag
\end{aligned} \tag{18}
$$

where $\Delta_{q,\hat{q}}(x,a,s) := q(x,a,s) - \hat{q}(x,a,s)$. Thus, we finally have

$$
\begin{aligned}
& n(\mathbb{V}_{\mathcal{D}}[\nabla_\theta \hat{V}_{\text{r-DR}}(\pi_\theta; D)] - \nabla_\theta \hat{V}_{\text{r}}(\pi_\theta; D)]) \\
&= \mathbb{E}_{p(x)\pi_0(a|x)p(s|x,a)}\left[\frac{\rho^2}{p(o|x)^2} w(x,a)^2 g_\theta(x,a)^2 \left(\Delta_{q,\hat{q}^{\neg s}}(x,a,s)^2 - \Delta_{q,\hat{q}}(x,a,s)^2\right)\right]
\end{aligned}
$$

□

C Additional Experiments

C.1 Detailed Synthetic Experiment Results

In this section, we provide more detailed experimental results for the comparisons conducted in Section 5. Specifically, we present results for additional baseline methods, including what we refer to as **r-DM** and **s-DM**. These are regression-based methods (also known as Direct Methods), which use regression estimates to predict expected target rewards based solely on logging data. The policy is then learned by choosing the action with the highest predicted reward in a deterministic manner. Note that r-DM trains its regression model on a partial dataset due to the use of target rewards, whereas s-DM trains on the full dataset but only uses secondary rewards for training.

For the experiment involving weight-tuned methods (experiments with varying β values), we also compare the values of parameters $\hat{\gamma}^*$ and γ^* used for HyPeR (Tuned $\hat{\gamma}^*$) and HyPeR (Optimal γ^*).

Results. Figures 6, 7, and 8 present additional experiments related to the research questions addressed in Section 5, including r-DM and s-DM. Specifically, Figure 6 evaluates the policy values under varying observation probabilities for the target rewards, Figure 7 evaluates the methods with different training sizes, and Figure 8 presents results for different degrees of target-secondary reward correlation. Across all experiments and policy values, we observe that r-DM and s-DM perform significantly worse than the corresponding IPS and DR methods due to high bias in the regression estimates.

Figure 9 shows results with varying true weights β in the combined objective. In this experiment, we additionally compare two more baseline methods: **HyPeR(Tuned $\hat{\gamma}^*$ w/o replacement)** and **DR with $F(s,r)$**. HyPeR(Tuned $\hat{\gamma}^*$ w/o replacement), unlike our tuning method, performs tuning of γ using sampling **without replacement**, i.e, without bootstrapping. We compare this baseline method to empirically show the effectiveness of our tuning method of leveraging the bootstrapping procedure. DR with $F(s,r)$ is another additional baseline that naively combines the target and secondly rewards in its policy gradient estimation; it uses Doubly Robust where the reward is defined as $F(s,r) = r$ when $o = 1$, and otherwise $F(s,r) = F(s)$ as in the method s-DR.

In Figure 9, we also observe that the additionally compared baseline methods all performed worse than the corresponding DR methods. The result also demonstrates that parameter tuning *with* replacement empirically outperforms tuning *without* replacement for a range of the true weight β values in the combined policy value objective. Moreover, by comparing HyPeR against the DR with $F(s,r)$ method, the results empirically demonstrate how we combine the target and secondary rewards is indeed crucial and that HyPeR does it effectively.

We also share the tuned $\hat{\gamma}^*$ values and compare them with the ground-truths in Table 2 below.

Table 2: Comparison of tuned $\hat{\gamma}^*$ without replacement, tuned $\hat{\gamma}^*$ with replacement, and the ground-truth γ^* values for varying β values (Experiment 4)

β	Mean $\hat{\gamma}^*$ w/o Replacement	Mean $\hat{\gamma}^*$ w/ Replacement	Mean γ^*
0.0	0.5595	0.3971	0.4788
0.2	0.6075	0.5762	0.5067
0.4	0.7390	0.7241	0.5542
0.6	0.8681	0.8198	0.6371
0.8	0.9019	0.8771	0.8014
1.0	1.0000	1.0000	1.0000

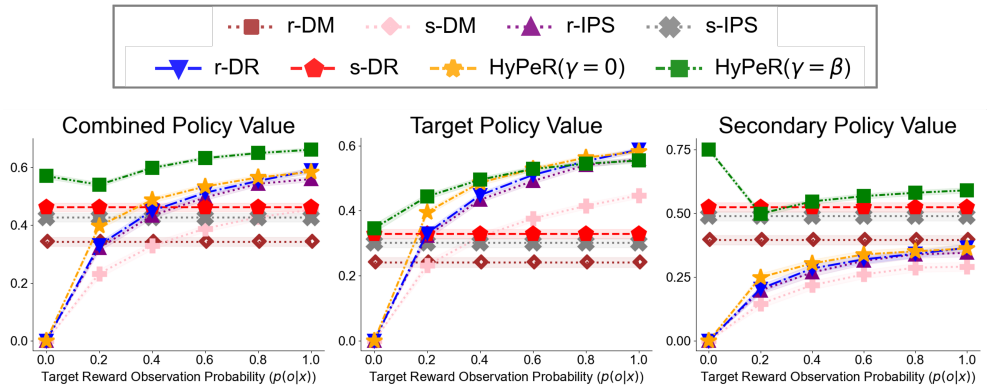

Figure 6: Comparing the combined, target, and secondary policy values of OPL methods with **varying target reward observation probabilities** ($p(o|x)$).

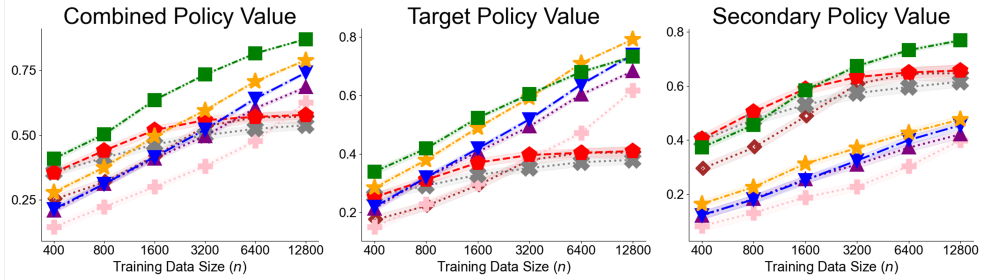

Figure 7: Comparing the combined, target, and secondary policy values of OPL methods with **varying training data sizes** (n)

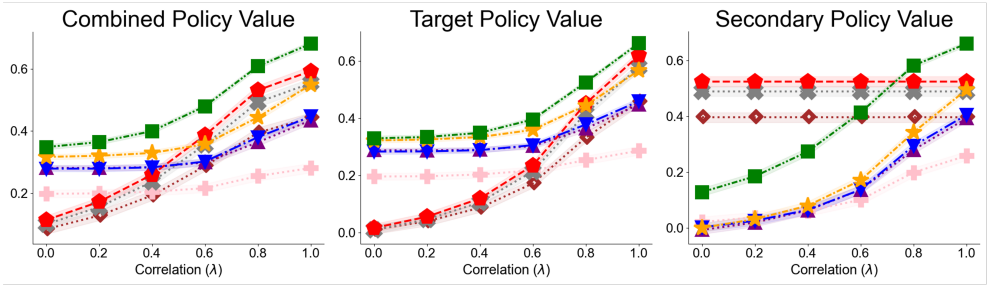

Figure 8: Comparing the combined, target, and secondary policy values of OPL methods with **varying degrees of target-secondary reward correlation** (λ). Secondary rewards can completely explain the target reward at $\lambda = 1$, and they are less correlated with smaller λ.

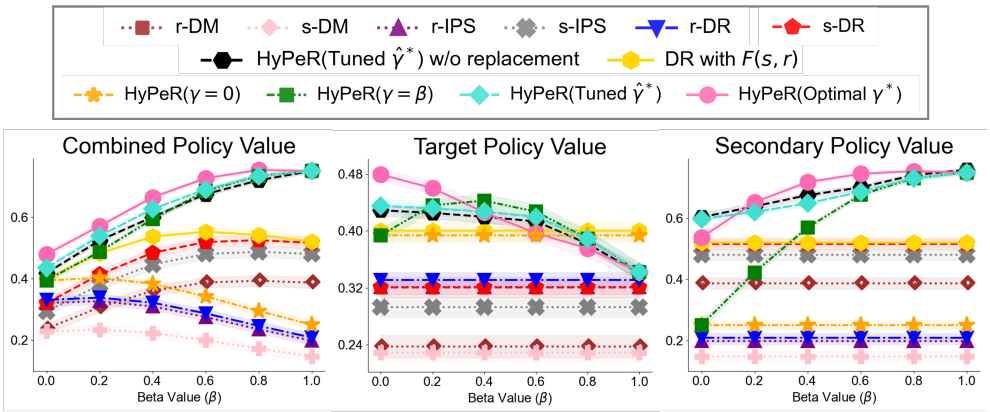

Figure 9: Comparing the combined, target, and secondary policy values of OPL methods with **varying** β, which is a weight that balances target and secondary policy values in the combined policy value. Higher β means the secondary policy value becomes more dominant.

C.2 Additional Synthetic Experiments

In this section, we additionally address three more research questions: two questions regarding the optimization performance under varying noise levels in each reward type and one concerning the effectiveness of HyPeR with estimated observation probabilities $p(o|x)$.

How does HyPeR perform under varying noise levels in secondary rewards? Figure 10 evaluates the relative policy value when the noise level in the secondary rewards is varied. As the secondary rewards become noisier, their variance increases. Consequently, methods that incorporate secondary rewards in their estimation (i.e., HyPeR($\gamma = \beta$), HyPeR($\gamma = 0$), s-DR, s-IPS, and s-DM) perform worse as noise increases. Most notably, the performance of s-IPS and r-DR degrades the most. In contrast, while HyPeR-based methods are also affected by the noise, their performance does not degrade as severely as that of the baseline methods that rely solely on secondary rewards, demonstrating its robustness due to the effective combination of the two types of rewards.

How does HyPeR perform under varying noise levels in target rewards? Figure 11 presents the results for different noise levels in the target rewards. As the noise increases, the variance in the target rewards becomes higher. This affects methods that utilize target rewards in their estimation (i.e., HyPeR($\gamma = \beta$), HyPeR($\gamma = 0$), r-DR, r-IPS, and r-DM). While HyPeR methods are negatively impacted, they consistently outperform the baseline methods that rely exclusively on target rewards, showing robustness as well.

How does HyPeR perform with estimated observation probabilities? To investigate the robustness of HyPeR to estimated probabilities $p(o|x)$, we conducted experiments under unknown $p(o|x)$ with varying levels of noise. The results are shown in Figure 12, where the methods that rely on $p(o|x)$ are affected by the estimation error and the presence of noise. We observed, however, that HyPeR consistently achieves better combined policy values compared to the baseline methods even under the realistic case with estimated probabilities $p(o|x)$. In particular, HyPeR (Tuned $\hat{\gamma}^*$) demonstrates better robustness to noise in $p(o|x)$, as it can adaptively assign greater weight (via the tuning procedure) to the secondary policy gradient when effective.

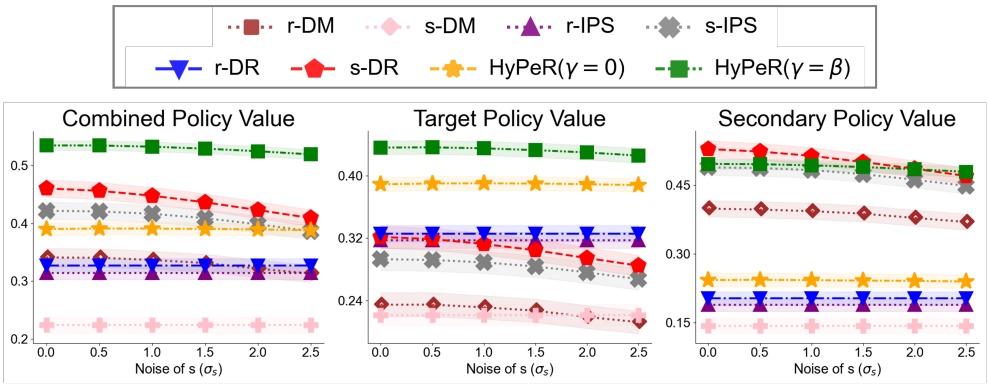

Figure 10: Comparing the combined, target, and secondary policy values of OPL methods with **varying noise of secondary reward** (σ_s).

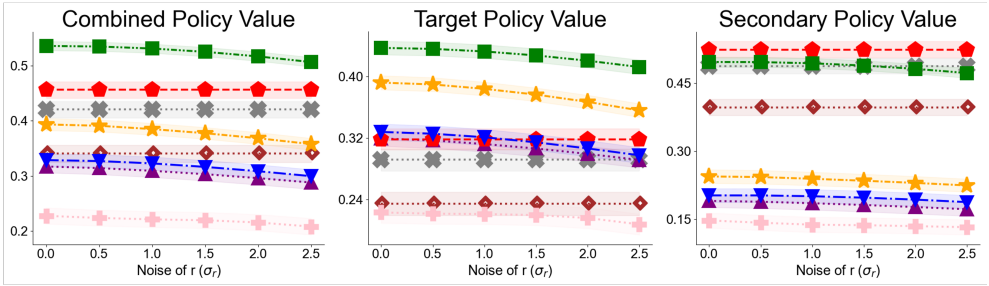

Figure 11: Comparing the combined, target, and secondary policy values of OPL methods with **varying noise of target reward** (σ_r).

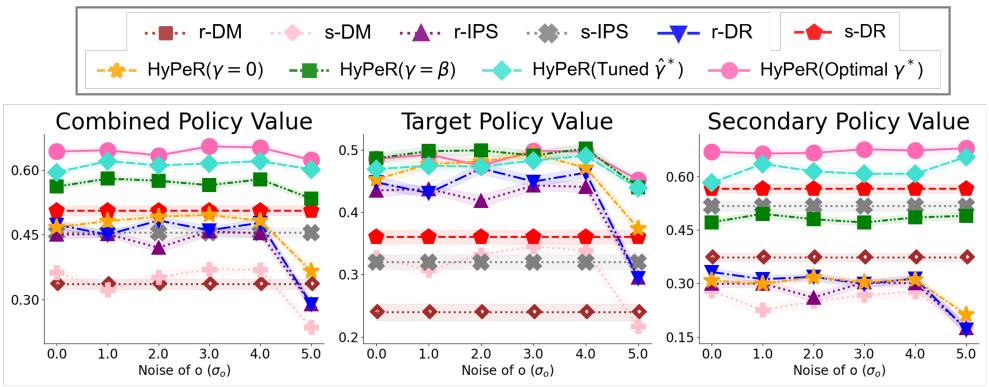

Figure 12: Comparing the combined, target, and secondary policy values of OPL methods with **varying noise in the observation probability** (σ_o). The probabilities for each user ($p(o|x)$) is fully unknown and requires estimation.

C.3 Detailed and Additional Real-world Experiment Results

In this section, we share more detailed experiment results of the real-world data experiments in Section 6. We add r-DM, r-IPS, s-DM, and s-IPS as baselines compared to the main text.

Results. Figures 13, 14, and 15 present more detailed results from the real-data experiments in Section 6, including additional methods for comparison. Figure 13 evaluates the policy values under varying observation probabilities of the target rewards, while Figure 14 evaluates the methods with different training data sizes. Figure 15 evaluates the methods with different true weights β in the objective. In this experiment, we also compare DR with $F(s, r)$ (the yellow line), which serves as a baseline that naively utilizes both target and secondary rewards when available.

In this section, we also evaluate the relative policy values when varying the size of the action set $|\mathcal{A}|$, as shown in Figure 16. As the action set size increases, the performance of all methods degrades due to an increase in variance in gradient estimation. However, we observe that HyPeR methods consistently outperform all other methods across all evaluated policy values.

Below, Tables 3, 4, 5, and 6 present the tuned $\hat{\gamma}^*$ values for the four real-world experiments, compared to the true optimal value γ^*, averaged over the number of simulations.

Table 3: Comparison of γ^* and $\hat{\gamma}^*$ values for Varying Target Reward Observation Probabilities ($p(o|x)$) (Corresponds to Figure 13)

$p(o\|x)$	Mean γ^* Value	Mean $\hat{\gamma}^*$ Value
0.0	1.0000	1.0000
0.2	0.6262	0.5926
0.4	0.5811	0.5542
0.6	0.5124	0.5002
0.8	0.4771	0.4701
1.0	0.4299	0.4512

Table 4: Comparison of γ^* and $\hat{\gamma}^*$ values with Varying Training Data Sizes (n) (Corresponds to Figure 14)

Data Size	Mean γ^* Value	Mean $\hat{\gamma}^*$ Value
500	0.7205	0.5458
1000	0.6612	0.5890
2000	0.5678	0.6392
4000	0.4462	0.5309
8000	0.3356	0.4794
16000	0.2878	0.4306

Table 5: Comparison of γ^* and $\hat{\gamma}^*$ values for Different Beta Values (β) (Corresponds to Figure 15)

β	Mean γ^* Value	Mean $\hat{\gamma}^*$ Value
0.0	0.5743	0.5287
0.2	0.6603	0.5833
0.4	0.7193	0.6386
0.6	0.7577	0.6549
0.8	0.8592	0.8021
1.0	1.0000	1.0000

Table 6: Comparison of $\hat{\gamma}^*$ and γ^* values for Different Action Sizes ($|\mathcal{A}|$) (Corresponds to Figure 16)

Action Size	Mean γ^* Value	Mean $\hat{\gamma}^*$ Value
25	0.5998	0.5312
50	0.6057	0.5159
100	0.5686	0.5008
200	0.6241	0.4925
400	0.5972	0.5328
800	0.6124	0.5086

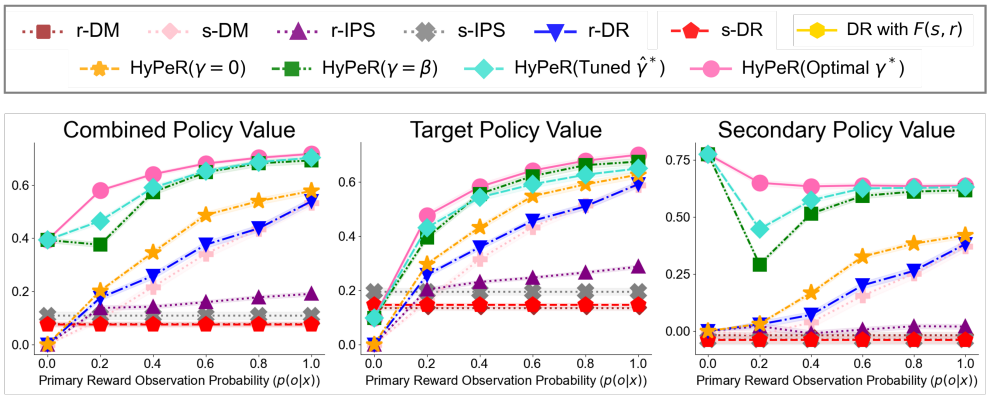

Figure 13: Comparing the combined, target, and secondary policy values of OPL methods with **varying target reward observation probabilities** ($p(o|x)$) on KuaiRec Dataset

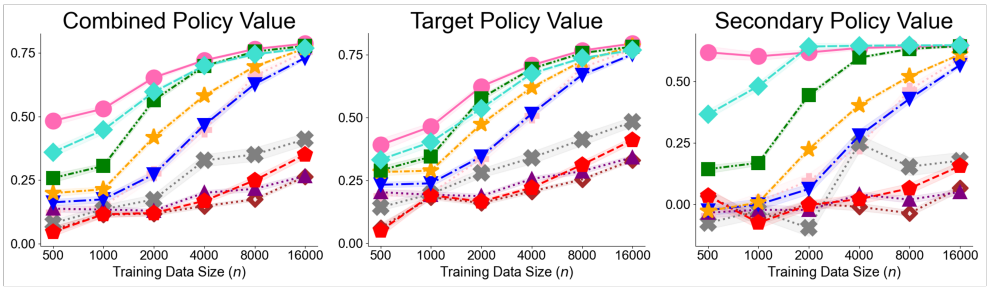

Figure 14: Comparing the combined, target, and secondary policy values of OPL methods with **varying training data sizes** (n) on KuaiRec Dataset

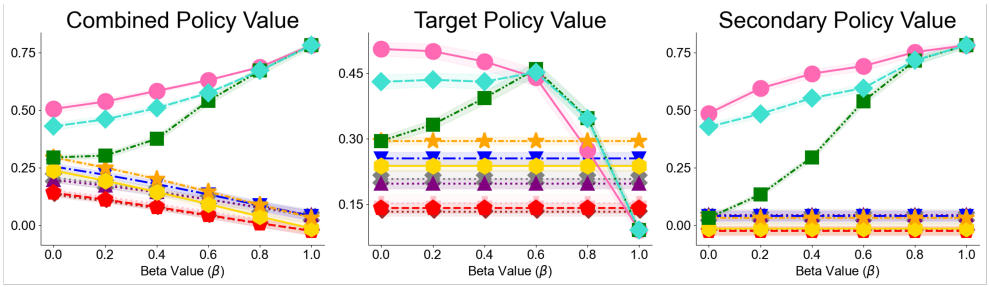

Figure 15: Comparing the combined, target, and secondary policy values of OPL methods with **varying** β, which is a weight that balances target and secondary policy values in the combined policy value. Higher β means the secondary policy value becomes more dominant on KuaiRec Dataset.

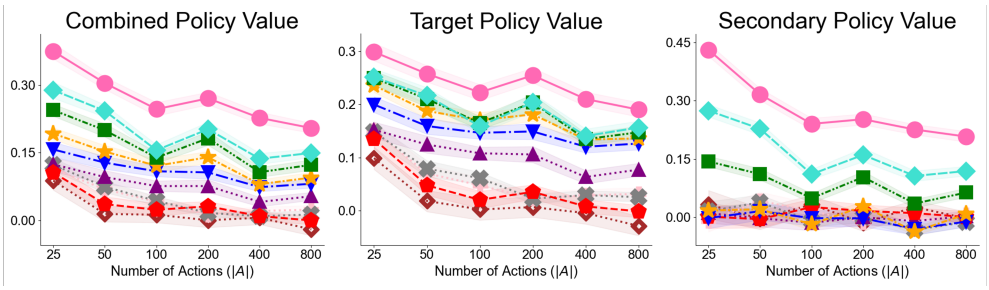

Figure 16: Comparing the combined, target, and secondary policy values of OPL methods with **varying action size** $|\mathcal{A}|$ on KuaiRec Dataset.