Offline Multi-Objective Bandits: From Logged Data to Pareto-Optimal Policies

Ji Cheng, Song Lai, Shunyu Yao, Bo Xue*

Department of Computer Science, City University of Hong Kong, Hong Kong SAR, China
The City University of Hong Kong Shenzhen Research Institute, Shenzhen, China
{J.Cheng, songlai2-c, shunyuyao8-c, boxue4-c}@my.cityu.edu.hk

Abstract

Offline policy learning from logged data is a critical paradigm for enabling effective decision-making without costly online exploration. However, its application has been largely confined to single-objective problems, a stark contrast to real-world scenarios where decision-making inherently involves navigating multiple, often conflicting, objectives. This paper introduces a comprehensive framework for Offline Multi-Objective Bandits (OffMOB), providing a principled solution to the fundamental challenge of learning Pareto-optimal policies from a static dataset. Our core contribution is a novel algorithm that uniquely integrates the pessimism principle with multi-objective optimization to safely learn from off-policy data. Crucially, our approach transcends the primary limitation of scalarization techniques, which are restricted to finding a single policy for a pre-defined preference. Instead, OffMOB directly approximates the entire Pareto front, learning a single, flexible policy model capable of generating an optimal action for any desired trade-off. To rigorously evaluate performance, we introduce the Tchebycheff sub-optimality metric and establish the first finite-sample generalization bounds for this problem class, proving that our algorithm converges to the true Pareto front under practical data coverage assumptions. Extensive experiments on complex benchmarks demonstrate that OffMOB significantly outperforms existing methods, identifying the complete set of optimal trade-offs where naive extensions and single-objective methods fail.

Code — https://github.com/jicheng9617/OffMOB

1 Introduction

Sequential decision-making from logged data is a central challenge in modern AI, with offline multi-armed bandits (MABs) providing a foundational framework for applications from recommender systems to personalized medicine (Slivkins et al. 2019; Lattimore and Szepesvári 2020; Dai et al. 2022). In the offline (or batch) setting, an agent learns a policy from a fixed dataset collected by a potentially unknown behavior policy, without any further online interaction. While significant progress has been made in offline learning for single-objective rewards (Brandfonbrener et al. 2021; Nguyen-Tang et al. 2022; Sakhi, Alquier, and Chopin 2023; Wang, Krishnamurthy, and Slivkins 2024; Liu et al. 2025), real-world problems rarely align with a single performance metric. For instance, a recommender system must balance user engagement with fairness and diversity (Coppolillo, Manco, and Gionis 2024), while a clinical treatment plan must maximize efficacy while minimizing side effects and costs (Kong, Xu, and Yang 2008). Optimizing for a simple scalar reward in such scenarios can lead to unintended and detrimental outcomes.

The multi-objective bandit framework addresses this by modeling rewards as vectors, where each component corresponds to a distinct objective (Drugan and Nowe 2013). The goal is not to find a single best-performing policy, but rather to learn a policy capable of generating the Pareto front: the set of all policies that offer optimal trade-offs (i.e., are not dominated by any other policy across all objectives) (Hayes et al. 2022). While online multi-objective bandits have been studied extensively, the critical offline setting remains largely unexplored. This gap is not incidental; it stems from a fundamental technical challenge. Standard offline methods fail because they are designed to estimate a single expected value. Extending them naively, for example by scalarizing the reward vector, requires pre-defined objective preferences (weights) which are often unavailable, and fails to uncover the full spectrum of possible solutions. Learning the entire Pareto front from off-policy data is a far harder problem, as estimation errors in the high-dimensional reward space can lead to a catastrophic collapse or misidentification of the optimal policy set.

In this paper, we bridge this gap by providing the first systematic study of Offline Multi-Objective Bandits (OffMOB). We tackle the core challenges of learning Pareto-optimal policies from logged contextual bandit data. Our work is centered on two fundamental questions: (i) Multi-Objective Counterfactual Estimation: *How can we reliably estimate the vector-valued outcomes of policies from biased, logged data, especially for actions that were rarely taken?* (ii) Pareto Front Learning: *How can we construct a policy that represents the entire set of Pareto optimal solutions from these noisy and offline estimates, while ensuring robustness against uncertainty?*

To address these challenges, we develop a comprehensive framework for OffMOB. Our main contributions are:

- We provide a rigorous formulation for the offline multi-

*Corresponding author.

objective contextual bandit problem. We define the concept of a preference-aware policy that can represent the entire Pareto front and introduce a new performance metric, the expected Tchebycheff (TCH) sub-optimality, to measure the quality of a learned policy against the true Pareto front across all possible trade-off preferences.

- We propose OffMOB, an algorithm designed specifically for this setting, which leverages neural networks to model the vector-valued reward functions and integrates the principle of pessimism into the scalarization framework. This ensures that policy evaluation is robust to estimation uncertainty, particularly for under-represented actions in the offline data.
- We establish the first finite-sample generalization bounds for OffMOB. Our analysis shows that our algorithm's sub-optimality is bounded under a practical multi-objective data coverage assumption, formally connecting data quality to the ability to recover the true Pareto front.
- We conduct experiments on synthetic data. Our results demonstrate that our proposed methods significantly outperform naive baselines, identifying complex Pareto fronts where simpler approaches fail. We highlight cases where ignoring the multi-objective structure leads to demonstrably suboptimal or even harmful policies.

2 Problem Setting

In this section, we formalize the problem of offline multi-objective contextual bandits. We extend the standard offline bandit framework to accommodate vector-valued rewards and define the objective of learning Pareto-optimal policies from a pre-collected dataset.

2.1 Offline Multi-Objective Contextual Bandits

We consider a stochastic K-armed contextual bandit with m objectives. The learning process is based on a static dataset, $\mathcal{D}_n = \{(\boldsymbol{x}_t, a_t, \boldsymbol{r}_t)\}_{t=1}^n$, collected a priori by a behavior policy μ. For each entry in the dataset:

- $\boldsymbol{x}_t := \{\boldsymbol{x}_{t,a} \in \mathbb{R}^d : a \in [K]\}$ is the full context observed at round t, sampled from an unknown context distribution ρ.
- $a_t \in [K]$ is the action taken by the behavior policy μ.
- $\boldsymbol{r}_t \in \mathbb{R}^m$ is the vector-valued reward received, where each component $r_{t,i}$ corresponds to the i-th objective.

The reward for each objective $i \in [m]$ is generated as:

$$r_{t,i} = h_i(\boldsymbol{x}_{t,a_t}) + \xi_{t,i}, \tag{1}$$

where $h_i : \mathbb{R}^d \to \mathbb{R}$ is an unknown reward function for the i-th objective, and $\xi_{t,i}$ is a conditionally R-subgaussian noise term.

In our multi-objective framework, a policy must balance multiple objectives according to a decision-maker's preferences. We consider preference-aware policies $\pi : \mathbb{R}^d \times \Delta^{m-1} \to [K]$, which map both the context $\boldsymbol{x}$ and a preference vector $\boldsymbol{\lambda}$ to a distribution over actions. The preference vector $\boldsymbol{\lambda} \in \Delta^{m-1}$ (the $(m-1)$-dimensional simplex where $\sum_{i=1}^m \lambda_i = 1$ and $\lambda_i \geq 0$) encodes the relative importance of each objective.

The expected vector-valued reward for a policy π given a context $\boldsymbol{x}$ and preference $\boldsymbol{\lambda}$ is:

$$\boldsymbol{v}^\pi(\boldsymbol{x}, \boldsymbol{\lambda}) = \mathbb{E}_{a\sim\pi(\cdot|\boldsymbol{x},\boldsymbol{\lambda})}[\boldsymbol{h}(\boldsymbol{x}_a)] \in \mathbb{R}^m, \tag{2}$$

where $\boldsymbol{h}(\boldsymbol{x}_a) = [h_1(\boldsymbol{x}_a), \ldots, h_m(\boldsymbol{x}_a)]^\top$.

Our goal is to learn a single preference-aware policy that can adapt to any preference vector, effectively representing the entire Pareto front. For a fixed context $\boldsymbol{x}$, different preference vectors $\boldsymbol{\lambda}$ should lead to different trade-offs among objectives, with the learned policy ideally selecting actions corresponding to different Pareto-optimal solutions.

Definition 1 (Pareto Dominance) *A reward vector $\boldsymbol{v}$ Pareto-dominates another reward vector $\boldsymbol{v}'$ (denoted $\boldsymbol{v} \succ \boldsymbol{v}'$) if $v_i \geq v_i'$ for all $i \in [m]$ and there exists $j \in [m]$ such that $v_j > v_j'$. A policy π^* is Pareto-optimal if its reward vector is not Pareto-dominated by any other policy's reward vector. The set of all such policies constitutes the Pareto front.*

2.2 Performance Evaluation via Tchebycheff Scalarization

To evaluate how well a preference-aware policy approximates the entire Pareto front, we need a metric that considers its performance across all possible preferences. We adopt the Tchebycheff scalarization approach, which converts the multi-objective problem into a scalar optimization problem for each preference vector.

For a given preference vector $\boldsymbol{\lambda}$, the Tchebycheff scalarization of a policy π given context $\boldsymbol{x}$ is defined as:

$$l^\pi(\boldsymbol{x}|\boldsymbol{\lambda}) = \max_{i\in[m]} \left\{\lambda_i \left(z_i^* - v_i^\pi(\boldsymbol{x}, \boldsymbol{\lambda})\right)\right\}, \tag{3}$$

where $z_i^* = \sup_\pi v_i^\pi(\boldsymbol{x}, \boldsymbol{\lambda})$ is the ideal point for objective i given context $\boldsymbol{x}$.

The Tchebycheff scalarization approach is theoretically grounded in its ability to recover the entire Pareto front:

Theorem 1 ((Choo and Atkins 1983)) *A policy π is weakly Pareto optimal if and only if there exists a weight vector $\boldsymbol{\lambda} \in \Delta^{m-1}$ such that π minimizes the Tchebycheff scalarization function $l^\pi(\boldsymbol{x}|\boldsymbol{\lambda})$.*

This theorem guarantees that by varying the preference vector $\boldsymbol{\lambda}$, we can in principle recover all Pareto-optimal policies. Therefore, a well-trained preference-aware policy should minimize $l^\pi(\boldsymbol{x}|\boldsymbol{\lambda})$ for each $\boldsymbol{\lambda}$, effectively approximating the entire Pareto front.

Definition 2 (Sub-optimality in Multi-Objective Setting) *For a preference-aware policy π, the sub-optimality measures the average performance gap across all possible preferences:*

$$\mathit{SubOpt}(\pi) = \mathbb{E}_{\boldsymbol{x}\sim\rho,\boldsymbol{\lambda}\sim\mathcal{U}(\Delta^{m-1})}\left[l^\pi(\boldsymbol{x}|\boldsymbol{\lambda}) - l^{\pi^*_{\boldsymbol{\lambda}}}(\boldsymbol{x}|\boldsymbol{\lambda})\right], \tag{4}$$

where $\pi^*_{\boldsymbol{\lambda}} = \arg\min_{\pi'} l^{\pi'}(\boldsymbol{x}|\boldsymbol{\lambda})$ is the optimal policy for the given preference vector $\boldsymbol{\lambda}$, and $\mathcal{U}(\Delta^{m-1})$ denotes the uniform distribution over the simplex.

This metric evaluates the expected gap between the learned policy and the optimal policy across all possible preference trade-offs and contexts. By taking the expectation over the entire simplex, we ensure that the learned policy must perform well for all preferences, not just a subset. A low sub-optimality indicates that the learned policy successfully approximates the entire Pareto front, providing near-optimal solutions for any preference vector a decision-maker might specify.

2.3 Neural Network Reward Function Approximation

To estimate the unknown reward functions h_i without prior knowledge of their parametric form, we model each one using a fully connected neural network (NN) and analyze its behavior within the Neural Tangent Kernel (NTK) framework (Jacot, Gabriel, and Hongler 2018).

For each objective $i \in [m]$, we use a dedicated NN with depth $L \geq 2$ and width m_{NN} per hidden layer. The architecture for the i-th reward function is defined as:

$$\hat{h}_i(\boldsymbol{x}; \boldsymbol{\theta}_i) = \mathbf{W}_L^i \cdot \sigma\left(\mathbf{W}_{L-1}^i \cdot \sigma(\cdots \sigma(\mathbf{W}_1^i \boldsymbol{x}))\right), \quad (5)$$

where $\sigma(z) = \max\{z, 0\}$ is the ReLU activation function, $\mathbf{W}_1^i \in \mathbb{R}^{m_{\text{NN}} \times d}$, $\mathbf{W}_\ell^i \in \mathbb{R}^{m_{\text{NN}} \times m_{\text{NN}}}$ for $2 \leq \ell < L$, and $\mathbf{W}_L^i \in \mathbb{R}^{1 \times m_{\text{NN}}}$. The parameter vector $\boldsymbol{\theta}_i$ contains all flattened weights for the i-th network.

In the overparameterized regime, where m_{NN} is sufficiently large, the NTK theory allows us to analyze the network's behavior. We consider a shared random initialization $\boldsymbol{\theta}_0$ for all networks. The network's output can be well-approximated by a linear function in its gradient feature space. We define the gradient feature mapping as: $g(\boldsymbol{x}; \boldsymbol{\theta}_0) = \nabla_{\boldsymbol{\theta}} \hat{h}(\boldsymbol{x}; \boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \in \mathbb{R}^p$, where $p = m_{\text{NN}}(d+1) + m_{\text{NN}}^2(L-2)$.

A key result from the NTK literature (Cao and Gu 2019; Gu et al. 2024) is that for a sufficiently wide network, there exists a parameter vector $\boldsymbol{\theta}_i^*$ such that the true unknown reward function h_i can be represented as:

$$h_i(\boldsymbol{x}) = \langle g(\boldsymbol{x}; \boldsymbol{\theta}_0), \boldsymbol{\theta}_i^* - \boldsymbol{\theta}_0 \rangle, \quad \forall \boldsymbol{x} \in \mathcal{X}. \quad (6)$$

This linearizes the problem, allowing us to estimate the unknown functions by learning the vectors $\boldsymbol{\theta}_i^*$ in the high-dimensional feature space defined by $g(\cdot; \boldsymbol{\theta}_0)$.

3 Learning Algorithm

In this section, we introduce OffMOB, a novel algorithm designed to learn a preference-aware policy from an offline dataset. The core principle of OffMOB is to combine neural network-based reward estimation with a pessimistic approach to policy evaluation, tailored for the multi-objective setting. This ensures that the learned policy is both robust to the limitations of the offline data and capable of approximating the entire Pareto front.

The OffMOB algorithm, detailed in Algorithm 1, operates in two primary stages at each step of processing the offline data: (1) learning the underlying reward functions for all objectives, and (2) constructing a pessimistic policy based on these learned functions and their associated uncertainty.

3.1 Neural Reward Estimation in NTK Space

The foundation of our approach is the accurate estimation of the unknown reward functions $\{h_i\}_{i=1}^m$. As outlined in Section 2.3, we model each h_i with a neural network and leverage the NTK framework to analyze its training dynamics.

The learning process for the network weights is performed via an online gradient descent update on a ridge regression objective, as shown in Line 8 of Algorithm 1. For each data point $(\boldsymbol{x}_t, a_t, \boldsymbol{r}_t)$ from the dataset $\mathcal{D}_n$, we update the weights $\boldsymbol{W}_i$ of the i-th network to minimize the following loss:

$$\mathcal{L}_t^i(\boldsymbol{W}) = \frac{1}{2}(f_{\boldsymbol{W}}(\boldsymbol{x}_{t,a_t}) - r_{t,i})^2 + \frac{m_{\text{NN}}\lambda_{\text{NN}}}{2}\|\boldsymbol{W} - \boldsymbol{W}_i^{(0)}\|_F^2. \quad (7)$$

This objective consists of two key components. The first term is the standard squared error, which drives the network's prediction $f_{\boldsymbol{W}}(\boldsymbol{x}_{t,a_t})$ towards the observed reward $r_{t,i}$. The second term is a regularization penalty that keeps the learned weights $\boldsymbol{W}$ close to their initial values $\boldsymbol{W}_i^{(0)}$. This specific form of regularization is crucial for the validity of the NTK approximation, ensuring that the network operates within the regime where its behavior can be accurately described by the linear model in the feature space induced by the gradient at initialization. The special initialization in Line 1 is a standard technique in NTK analysis to ensure the resulting kernel is well-conditioned and the network has sufficient expressive power.

3.2 Pessimistic Policy Construction with Confidence Bounds

A key challenge in offline reinforcement learning is mitigating distributional shift; the learned policy may favor actions that were rarely taken by the behavior policy, leading to unreliable value estimates. To address this, OffMOB adopts the principle of pessimism in the face of uncertainty. Instead of optimistically exploring as in online settings, our policy conservatively evaluates actions based on a lower confidence bound (LCB) of their estimated rewards.

As detailed in Line 6 of Algorithm 1, the policy $\hat{\pi}_t$ for a given preference $\boldsymbol{\lambda}$ is constructed by selecting the action that minimizes a pessimistic estimate of the Tchebycheff scalarization. The objective for this minimization is:

$$\mathcal{L}_t(\boldsymbol{u}, \boldsymbol{\lambda}) = \max_{i \in [m]} \lambda_i \left(z_i^* - \hat{f}_t^i(\boldsymbol{u})\right), \quad (8)$$

where the lower confidence bound for the reward of objective i is defined as:

$$\hat{f}_t^i(\boldsymbol{u}) = f_{\boldsymbol{W}_i^{(t-1)}}(\boldsymbol{u}) - \beta_{t-1}\|\nabla f_{\boldsymbol{W}_i^{(t-1)}}(\boldsymbol{u}) \cdot m_{\text{NN}}^{-\frac{1}{2}}\|_{(\boldsymbol{\Lambda}_{t-1}^i)^{-1}} \quad (9)$$

Here, the term subtracted from the mean prediction $f_{\boldsymbol{W}_i^{(t-1)}}(\boldsymbol{u})$ represents the uncertainty. The matrix $\boldsymbol{\Lambda}_t^i$, updated in Line 7, accumulates the outer products of the gradient features. It serves as a covariance matrix in the NTK feature space, capturing the amount of information we have gathered for different contexts. The norm $\|\cdot\|_{(\boldsymbol{\Lambda}_{t-1}^i)^{-1}}$ is

Algorithm 1: OffMOB: Offline Multi-Objective Bandit Algorithm

Require: Offline data $\mathcal{D}_n = \{(\boldsymbol{x}_t, a_t, \boldsymbol{r}_t)\}_{t=1}^n$, step sizes $\{\eta_t\}_{t=1}^n$, regularization parameter $\lambda_{\text{NN}} > 0$, confidence parameters $\{\beta_t\}_{t=1}^n$, number of preference samples B_λ.

1: Initialize $\boldsymbol{W}_i^{(0)}$ for each objective $i \in [m]$ as follows: set $\boldsymbol{W}_{i,\ell}^{(0)} = [\bar{\boldsymbol{W}}_\ell, \boldsymbol{0}; \boldsymbol{0}, \bar{\boldsymbol{W}}_\ell], \forall \ell \in [L-1]$ where each entry of $\bar{\boldsymbol{W}}_\ell$ is generated independently from $\mathcal{N}(0, 4/m_{\text{NN}})$, and set $\boldsymbol{W}_{i,L}^{(0)} = [\boldsymbol{w}^T, -\boldsymbol{w}^T]$ where each entry of $\boldsymbol{w}$ is generated independently from $\mathcal{N}(0, 2/m_{\text{NN}})$.
2: $\boldsymbol{\Lambda}_0^i \leftarrow \lambda_{\text{NN}} \boldsymbol{I}$ for all $i \in [m]$.
3: **for** $t = 1, \ldots, n$ **do**
4: Retrieve $(\boldsymbol{x}_t, a_t, \boldsymbol{r}_t)$ from $\mathcal{D}_n$.
5: Sample B_λ preference vectors $\{\boldsymbol{\lambda}_b\}_{b=1}^{B_\lambda}$ uniformly from Δ^{m-1}.
6: $\hat{\pi}_t(\boldsymbol{x}|\boldsymbol{\lambda}_b) \leftarrow \arg\min_{a \in [K]} L_t(\boldsymbol{x}_a, \boldsymbol{\lambda}_b)$, for all $\boldsymbol{x} = \{\boldsymbol{x}_a \in \mathbb{R}^d : a \in [K]\}$ and $\boldsymbol{\lambda}_b$, where $L_t(\boldsymbol{u}, \boldsymbol{\lambda}) = \max_{i \in [m]} \lambda_i \left(z_i^* - \left(f_{\boldsymbol{W}_i^{(t-1)}}(\boldsymbol{u}) - \beta_{t-1} \|\nabla f_{\boldsymbol{W}_i^{(t-1)}}(\boldsymbol{u}) \cdot m_{\text{NN}}^{-1/2}\|_{(\boldsymbol{\Lambda}_{t-1}^i)^{-1}} \right) \right)$
7: $\boldsymbol{\Lambda}_t^i \leftarrow \boldsymbol{\Lambda}_{t-1}^i + \text{vec}(\nabla f_{\boldsymbol{W}_i^{(t-1)}}(\boldsymbol{x}_{t,a_t})) \cdot \text{vec}(\nabla f_{\boldsymbol{W}_i^{(t-1)}}(\boldsymbol{x}_{t,a_t}))^T / m_{\text{NN}}, \forall i \in [m]$.
8: $\boldsymbol{W}_i^{(t)} \leftarrow \boldsymbol{W}_i^{(t-1)} - \eta_t \nabla \mathcal{L}_t^i(\boldsymbol{W}_i^{(t-1)}), \forall i \in [m]$, where $\mathcal{L}_t^i(\boldsymbol{W}) = \frac{1}{2}(f_{\boldsymbol{W}}(\boldsymbol{x}_{t,a_t}) - r_{t,i})^2 + \frac{m_{\text{NN}} \lambda_{\text{NN}}}{2} \|\boldsymbol{W} - \boldsymbol{W}_i^{(0)}\|_F^2$.
9: **end for**

Output: For a given preference $\boldsymbol{\lambda}$, return policy $\hat{\pi}(\cdot|\boldsymbol{x}, \boldsymbol{\lambda})$ computed using the final parameters $\{\boldsymbol{W}_i^{(n)}\}_{i=1}^m$.

thus large for context-action pairs whose features are dissimilar to those seen in the data, resulting in a lower, more pessimistic reward estimate. The hyperparameter β_{t-1} controls the level of pessimism. By using this pessimistic Tchebycheff value, the policy avoids actions with high uncertainty, ensuring that its decisions are grounded in sufficient evidence from the offline dataset.

3.3 Approximating the Full Pareto Front

The ultimate goal of OffMOB is to learn a single, unified policy that can cater to any user preference $\boldsymbol{\lambda} \in \Delta^{m-1}$. The algorithm achieves this by integrating the preference vector directly into the policy construction while learning the objective-specific reward functions independently.

The critical step for achieving this is Line 5 of Algorithm 1, where we sample a batch of preference vectors $\boldsymbol{\lambda}_b$ from the simplex. Although these sampled preferences are used to define the target policy $\hat{\pi}_t$ in Line 6, the actual model updates in Lines 7 and 8 are performed independently for each objective i. This design decouples the learning of the world dynamics (the reward functions h_i) from the application of user preferences.

As a result, the final set of learned neural networks $\{\boldsymbol{W}_i^{(n)}\}_{i=1}^m$ implicitly encodes the reward functions for all objectives. At deployment, a decision-maker can provide any preference vector $\boldsymbol{\lambda}$, and the policy $\hat{\pi}(\cdot|\boldsymbol{x}, \boldsymbol{\lambda})$ can instantly compute the corresponding optimal action by solving the pessimistic Tchebycheff minimization problem. This mechanism allows OffMOB to effectively represent and approximate the entire Pareto front with a single model.

4 Generalization Analysis

In this section, we provide a theoretical analysis of the OffMOB algorithm. Our goal is to derive an upper bound for the multi-objective sub-optimality metric, SubOpt(π), defined in Section 2.2. The analysis leverages the properties of the NTK to handle the non-linear reward function approximation and introduces a novel data coverage assumption tailored for the offline multi-objective setting.

Our analysis is grounded in the NTK framework (Jacot, Gabriel, and Hongler 2018), which characterizes the behavior of infinitely wide neural networks. The NTK allows us to analyze the complex dynamics of neural network training through the lens of a fixed kernel.

Definition 3 ((Jacot, Gabriel, and Hongler 2018)) *Let $\{\boldsymbol{x}^{(i)}\}_{i=1}^{nK} = \{\boldsymbol{x}_{t,a} \in \mathbb{R}^d : t \in [n], a \in [K]\}$ be the set of all possible context-action pairs from the dataset. The NTK matrix $\mathbf{H} \in \mathbb{R}^{nK \times nK}$ is defined recursively. Let $\tilde{\mathbf{H}}_{i,j}^{(1)} = \boldsymbol{\Sigma}_{i,j}^{(1)} = \langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle$. For layers $\ell = 1, \ldots, L-1$, define:*

$$\mathbf{A}_{i,j}^{(\ell)} = \begin{bmatrix} \boldsymbol{\Sigma}_{i,i}^{(\ell)} & \boldsymbol{\Sigma}_{i,j}^{(\ell)} \\ \boldsymbol{\Sigma}_{j,i}^{(\ell)} & \boldsymbol{\Sigma}_{j,j}^{(\ell)} \end{bmatrix},$$

$$\boldsymbol{\Sigma}_{i,j}^{(\ell+1)} = 2\mathbb{E}_{(u,v) \sim \mathcal{N}(\mathbf{0}, \mathbf{A}_{i,j}^{(\ell)})} [\sigma(u)\sigma(v)],$$

$$\widetilde{\mathbf{H}}_{i,j}^{(\ell+1)} = 2\widetilde{\mathbf{H}}_{i,j}^{(\ell)} \mathbb{E}_{(u,v) \sim \mathcal{N}(\mathbf{0}, \mathbf{A}_{i,j}^{(\ell)})} [\sigma'(u)\sigma'(z)] + \boldsymbol{\Sigma}_{i,j}^{(\ell+1)}.$$

The final NTK matrix is given by $\mathbf{H} = (\widetilde{\mathbf{H}}^{(L)} + \boldsymbol{\Sigma}^{(L)})/2$.

The NTK matrix $\mathbf{H}$ captures the inner products of the gradient feature vectors for all context-action pairs, effectively defining a geometry over the input space. The complexity of the function class that the neural network can represent on the given data can be characterized by the effective dimension of this kernel matrix (Valko et al. 2013; Zhou, Li, and Gu 2020).

Definition 4 (Effective Dimension) *Given the NTK matrix $\mathbf{H}$ and a regularization parameter $\lambda_{\text{NN}} > 0$, the effective dimension $\tilde{d}$ is defined as:*

$$\tilde{d} = \frac{\log \det(\mathbf{I} + \mathbf{H}/\lambda_{\text{NN}})}{\log(1 + nK/\lambda_{\text{NN}})}. \tag{10}$$

The effective dimension $\tilde{d}$ measures how quickly the eigenvalues of the Gram matrix $\mathbf{H}$ decay. It can be significantly smaller than the ambient dimension nK, especially if the eigenvalues exhibit a fast decay (e.g., polynomial or exponential), and it often grows only logarithmically with the number of data points n.

We now introduce the assumptions required for our analysis, extending standard assumptions from the single-objective NTK and offline bandit literature to our multi-objective context. The first assumption ensures the NTK matrix is well-behaved.

Assumption 1 (NTK Regularity and Input Structure) *There exists a constant $\lambda_0 > 0$ such that the NTK matrix $\mathbf{H} \succeq \lambda_0 \mathbf{I}$. Furthermore, for any context-action feature vector $\boldsymbol{x}_{t,a}$, we assume $\|\boldsymbol{x}_{t,a}\|_2 = 1$ and $[\boldsymbol{x}_{t,a}]_j = [\boldsymbol{x}_{t,a}]_{j+d/2}$ for all $j \in [d/2]$.*

The positive definiteness of $\mathbf{H}$ is a standard non-singularity condition in kernel and NTK literature (Arora et al. 2019; Zhou, Li, and Gu 2020), ensuring that the learned function is well-defined. The structural conditions on the input vectors are mild technical requirements for the theoretical analysis, which guarantee that the network output at initialization is zero, simplifying the subsequent regret bounds (Zhou, Li, and Gu 2020).

Our second key assumption concerns the data coverage of the offline dataset $\mathcal{D}_n$. In the offline setting, we cannot actively explore, so the quality of the learned policy depends entirely on the data collected by the behavior policy μ. A common, but often restrictive, assumption is uniform coverage, which requires μ to explore all actions sufficiently. We relax this significantly by extending the concept of empirical single-policy concentration (eSPC) (Rashidinejad et al. 2021) to the multi-objective domain.

Assumption 2 (MO-eSPC) *Let $\pi^*_{\boldsymbol{\lambda}} = \arg\min_{\pi'} g^{\pi'}(\boldsymbol{x}|\boldsymbol{\lambda})$ be the optimal policy for a given preference vector $\boldsymbol{\lambda}$. We assume there exists a constant $\kappa \in (0, \infty)$ such that for all $t \in [n]$:*

$$\sup_{\boldsymbol{\lambda} \in \Delta^{m-1}} \left\| \frac{\pi^*_{\boldsymbol{\lambda}}(\cdot|\boldsymbol{x}_t)}{\mu(\cdot|\mathcal{D}_{t-1}, \boldsymbol{x}_t)} \right\|_\infty \leq \kappa. \tag{11}$$

This assumption requires the behavior policy μ to provide sufficient coverage only over the set of Pareto-optimal policies, indexed by all possible preference vectors $\boldsymbol{\lambda}$. Unlike uniform coverage assumptions that demand exploration over all possible policies, MO-eSPC only requires that for any trade-off preference, the corresponding optimal action has a non-trivial probability of being selected by μ. This is a far more practical condition, as it allows the behavior policy to have been goal-oriented itself, as long as its goals were sufficiently diverse to cover the Pareto front. It also accommodates non-stationary behavior policies, such as those used in active data collection scenarios.

We now present the main theoretical result of this paper: an upper bound on the sub-optimality of the policy $\hat{\pi}$ learned by the OffMOB algorithm. The bound characterizes the policy's performance as a function of the number of data points n, the number of objectives m, the complexity of the problem measured by the effective dimension $\tilde{d}$, and the quality of the offline data quantified by the MO-eSPC coefficient κ.

Theorem 2 (Sub-optimality Bound for OffMOB) *Assume Assumptions 1 and 2 hold. For any $\delta \in (0, 1)$, there exist sufficiently large polynomial dependencies of the network width m_{NN} on $(n, K, L, \lambda_{\text{NN}}^{-1}, \lambda_0^{-1}, \log(m/\delta))$ such that with probability at least $1 - \delta$, the final policy $\hat{\pi}$ returned by Algorithm 1 satisfies:*

$$\textit{SubOpt}(\hat{\pi}) = O\left(m\kappa \sqrt{\frac{\tilde{d}\log(nK/\lambda_{\text{NN}})}{n}} + \sqrt{\frac{\log(1/\delta)}{n}} \right). \tag{12}$$

This theorem establishes that the sub-optimality of OffMOB converges to zero as the size of the offline dataset n increases. The bound scales linearly with the data coverage coefficient κ and with the number of objectives m. Importantly, the dependence on the number of data points and actions (n, K) is captured by the effective dimension $\tilde{d}$, which is typically much smaller than the ambient dimension, thus avoiding a vacuous bound.

We highlight several key aspects of this result. First, this theorem provides the first provably efficient generalization guarantee for offline multi-objective bandits using overparameterized neural networks. A critical feature is that the bound scales with the effective dimension $\tilde{d}$, which often grows only logarithmically with the sample size n. This ensures the bound is non-vacuous and meaningful for modern deep learning models. Second, the linear dependence on the coefficient κ demonstrates that OffMOB succeeds under a practical, concentrated data coverage assumption, avoiding the need for restrictive uniform exploration. These properties establish OffMOB as a theoretically sound and practical algorithm for this challenging setting.

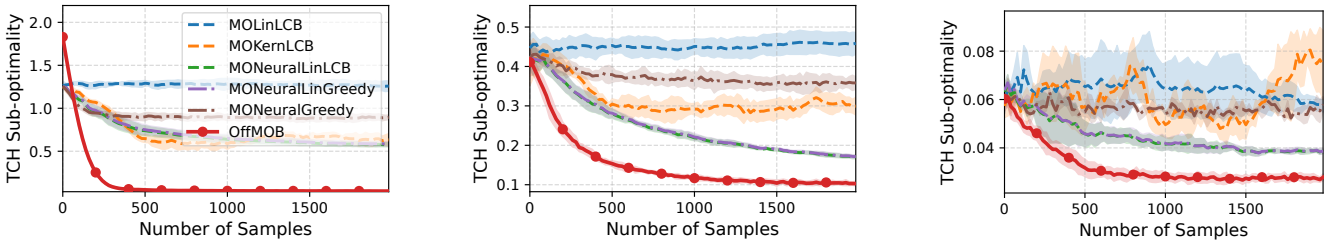
5 Experimental Results

In this section, we conduct experiments on synthetic datasets to evaluate the empirical performance of our proposed algorithm, OffMOB. We compare it with several representative baselines adapted for the offline multi-objective setting.

5.1 Experimental Setup

Baselines. We compare OffMOB with five baseline methods, each adapted to the multi-objective framework by using the Tchebycheff scalarization for decision-making.

- **MOLinLCB**: An extension of LinLCB (Jin, Yang, and Wang 2020) to the multi-objective setting. It models each of the m reward functions with a separate linear model and uses a pessimistic (LCB) approach for policy selection based on the Tchebycheff scalarization.
- **MOKernLCB**: An offline multi-objective counterpart of KernelUCB (Valko et al. 2013). It models each reward function within a Reproducing Kernel Hilbert Space (RKHS) and employs a pessimistic LCB strategy.
- **MONeuralLinLCB**: This baseline first uses the gradient of an initialized neural network, $\phi(\boldsymbol{x}_a) = \text{vec}(\nabla f_{\boldsymbol{W}^{(0)}}(\boldsymbol{x}_a))$, as a fixed feature extractor. It then



(a) $h_i(\boldsymbol{u}) = 10(\boldsymbol{u}^T \boldsymbol{a}_i)^2, \forall i \in [m]$ (b) $h_i(\boldsymbol{u}) = \boldsymbol{u}^T \boldsymbol{A}_i^T \boldsymbol{A}_i \boldsymbol{u}$ (c) $h_i(\boldsymbol{u}) = \cos(3\boldsymbol{u}^T \boldsymbol{c}_i)$

Figure 1: The Tchebycheff sub-optimality of offMOB versus the baseline algorithms on three types of nonlinear datasets.

applies MOLinLCB on these high-dimensional features. The network $f_{\boldsymbol{W}^{(0)}}$ is identical to the one used by OffMOB at initialization.

- **MONeuralLinGreedy**: Identical to MONeuralLinLCB but uses a greedy strategy. It makes decisions based on the empirical point estimates of the rewards, without considering uncertainty.
- **MONeuralGreedy**: This is an ablation of our OffMOB algorithm. It is identical to OffMOB in its use of trainable neural networks to model rewards, but it makes decisions greedily based on the network's current predictions, ignoring the pessimistic confidence bounds.

Evaluation Metric. We evaluate all algorithms using the expected Tchebycheff sub-optimality (SubOpt(π)) as defined in Section 2.2. For each experimental run, we approximate this metric by drawing a large number of test contexts from the true context distribution ρ and a large number of preference vectors $\boldsymbol{\lambda}$ from the uniform distribution over the simplex Δ^{m-1}.

Experiment Protocol. For each experimental setting, we vary the number of offline samples n and repeat each experiment for 5 independent runs. We report the mean sub-optimality and the corresponding 95% confidence intervals.

5.2 Implementation Details

Approximation. To manage computational complexity, we adopt standard approximation techniques. For OffMOB and MONeuralLinLCB, the covariance matrix $\boldsymbol{\Lambda}_t$ can be very large. We approximate it by its diagonal, which is a common and effective heuristic (Riquelme, Tucker, and Snoek 2018; Zhou, Li, and Gu 2020). For MOKernLCB, which scales cubically with the sample size, we fit the kernel model on the first 1,000 samples and use the resulting fixed model for evaluation if the dataset size exceeds this limit.

Data Generation. The offline dataset $\mathcal{D}_n$ is generated by a fixed behavior policy μ. We use a ϵ-greedy policy, and randomly select actions from the Pareto optima at each round. This simulates a scenario where the data is collected by a sub-optimal but reasonable goal-oriented policy. We set $\epsilon = 0.1$ for all experiments.

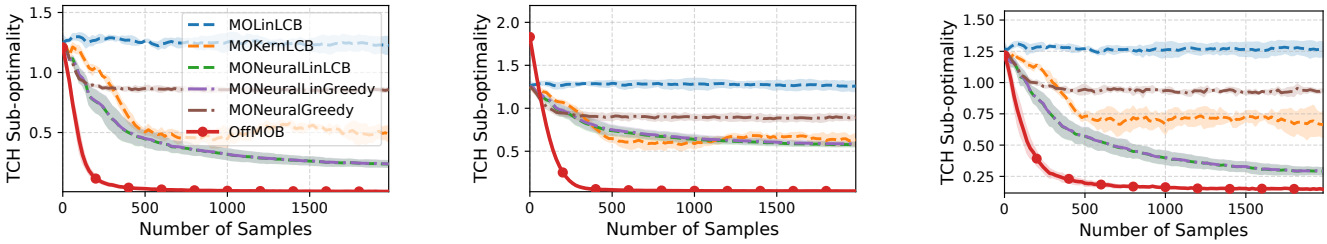
Hyperparameters. We set the regularization parameter $\lambda_{\text{NN}} = 0.1$ for all algorithms. For all LCB-based methods (OffMOB, MOLinLCB, MOKernLCB, MONeuralLinLCB), we perform a grid search for the uncertainty parameter β over the set $\{0.01, 0.05, 0.1, 1, 5, 10\}$. For MOKernLCB, we use the Radius Basis Function (RBF) kernel and search for its bandwidth parameter σ in $\{0.1, 1, 10\}$. For algorithms using trainable neural networks (OffMOB and MONeuralGreedy), we use the Adam optimizer (Kingma and Ba 2014) with a learning rate η searched over $\{0.0001, 0.001\}$ and an l_2-regularization parameter of 0.0001. For all neural network-based methods, we use a 2-layer MLP with hidden layer width of m_{NN}.

Synthetic Datasets We evaluate the algorithms on contextual bandit problems with synthetic non-linear reward functions, extended to a multi-objective scenario ($m = 3$). The reward functions are inspired by those in (Nguyen-Tang et al. 2022): $h_i(\boldsymbol{u}) = 10(\boldsymbol{u}^T \boldsymbol{a}_i)^2$, $h_i(\boldsymbol{u}) = \boldsymbol{u}^T \boldsymbol{A}_i^T \boldsymbol{A}_i \boldsymbol{u}$, and $h_i(\boldsymbol{u}) = \cos(3\boldsymbol{u}^T \boldsymbol{c}_i)$. Here, $\boldsymbol{a}_i, \boldsymbol{c}_i \in \mathbb{R}^d$ are random vectors on the unit sphere, and $\boldsymbol{A}_i \in \mathbb{R}^{d \times d}$ is random matrices with entries drawn from $\mathcal{N}(0, 1)$. The observed reward is $\boldsymbol{r}_t = \boldsymbol{h}_i(\boldsymbol{x}_{t,a_t}) + \boldsymbol{\xi}_t$, where $\boldsymbol{\xi}_t \sim \mathcal{N}(\boldsymbol{0}, \sigma \mathbf{I})$. For all instances, the context dimension is $d = 20$, the number of actions is $K = 30$, and the neural network width is $m_{\text{NN}} = 50$. The maximum number of offline samples is $n = 2,000$.

5.3 Results

Figure 1 shows that OffMOB consistently outperforms all baselines across three distinct non-linear datasets. The significant performance gap between OffMOB and its greedy counterpart, MONeuralGreedy, highlights the critical importance of incorporating pessimism to mitigate exploitation risk. Furthermore, the nearly identical performance of MONeuralLinLCB and MONeuralLinGreedy reveals that pessimism is ineffective when the underlying model is misspecified, as the resulting uncertainty estimates are unreliable. This underscores the necessity of OffMOB's expressive, trainable model for effective offline decision-making.

We further evaluate our algorithm to varying levels of observational noise. Figure 2 illustrates the performance of all methods on datasets with noise scales σ set to 0.01, 0.1, and 0.5. The results clearly show that OffMOB consistently maintains its superior performance over all baselines across these different noise conditions. While, as expected, the absolute sub-optimality for all algorithms increases with the noise level, OffMOB's performance degrades gracefully. It



(a) $\sigma = 0.01$ (b) $\sigma = 0.1$ (c) $\sigma = 0.5$

Figure 2: Comparison between different scales of the noise on quadratic reward function $h_i(\boldsymbol{u}) = 10(\boldsymbol{u}^T \boldsymbol{a}_i)^2$.

consistently retains a significant advantage over the next-best methods, even in the high-noise setting. This demonstrates the robustness of our approach in effectively learning reliable policies from stochastic reward signals.

6 Related Work

Offline Bandits and Policy Learning. Offline contextual bandits have become a central paradigm for decision-making from logged data, with applications in recommendation systems, healthcare, and digital marketing (Swaminathan and Joachims 2015; Gottesman et al. 2019; Bottou et al. 2013). In the offline setting, a learner must optimize policies solely from historical data generated by an unknown behavior policy, without the ability to interactively explore the environment (Langford and Zhang 2007; Dudík, Langford, and Li 2011). Most existing offline bandit algorithms focus on single-objective reward learning, utilizing techniques such as inverse propensity weighting (Strehl et al. 2010), doubly robust estimation (Wang, Agarwal, and Dudık 2017), and pessimism principles (Chen et al. 2021) to address the challenges of distributional shift and counterfactual estimation. Recent advances have extended these methods to high-capacity models, including neural networks, with theoretical guarantees under sufficient data coverage assumptions (Uehara and Sun 2022; Rashidinejad et al. 2021; Zhan, Zhu, and Xu 2021). However, these approaches are typically designed for scalar reward functions and do not address the unique challenges of multi-objective optimization.

Multi-Objective Bandits. Multi-objective bandit problems generalize the classic MAB framework by associating each action with a vector of rewards, corresponding to different, potentially conflicting objectives (Drugan and Nowe 2013; Tekin and Turğay 2018; Xue et al. 2024). The core challenge is to balance exploration and exploitation across objectives, aiming to identify Pareto-efficient actions or policies (Cheng et al. 2024, 2025). Drugan and Nowe (2013) introduced the Multi-Objective Multi-Armed Bandit (MOMAB) framework using Pareto dominance, followed by algorithms that analyze Pareto regret and trade-off surfaces (Xu and Klabjan 2023). Contextual and generalized linear extensions have also been studied (Turgay, Oner, and Tekin 2018; Lu et al. 2019; Xue et al. 2023), enabling more expressive models for user preferences. However, the majority of these methods operate in the online setting, where policies can interactively explore the environment, and typically assume access to instantaneous multi-objective feedback for all chosen actions.

Offline Multi-Objective Policy Learning and Pareto Regret. Despite the practical significance of multi-objective decision-making from logged data, the study of offline multi-objective bandits remains largely unexplored. Most offline learning research has focused on single-objective settings, with only limited attention to vector-valued outcomes in observational studies or batch reinforcement learning (Levine et al. 2020; Ishfaq et al. 2024). In the broader reinforcement learning context, some works have considered offline policy evaluation or improvement using scalarization techniques (Yang, Sun, and Narasimhan 2019; Abels et al. 2019). The challenge of learning policies that are Pareto-optimal, that is, not dominated in any objective, has motivated the use of Pareto regret as a key evaluation metric (Auer et al. 2016; Xu and Klabjan 2023). While recent advances in online multi-objective bandits have developed algorithms and analyzed Pareto regret under various structural assumptions (Lu et al. 2019; Cheng et al. 2024; Xue et al. 2025), these approaches fundamentally rely on active exploration and cannot be directly applied in the offline context, where only logged actions and their corresponding multi-objective rewards are available. As a result, the interplay between data coverage, counterfactual estimation, and Pareto policy learning in offline multi-objective bandits remains an open and important challenge.

7 Conclusion and Discussion

In this work, we established a principled framework for offline multi-objective contextual bandits. Our algorithm, OffMOB, effectively combines neural network approximators with the pessimism principle to learn a robust policy that can represent the entire Pareto front. We provided the first finite-sample sub-optimality guarantees for this setting and demonstrated empirically that OffMOB significantly outperforms relevant baselines, highlighting the importance of its design.

A significant extension would be to generalize our framework to the full offline multi-objective reinforcement learning setting. Finally, exploring methods to incorporate partial or uncertain user preferences could further enhance the practical utility of our approach.

Acknowledgments

The work described in this paper was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China [GRF Project No. CityU11215622].

References

Abels, A.; Roijers, D.; Lenaerts, T.; Nowé, A.; and Steckelmacher, D. 2019. Dynamic weights in multi-objective deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, 11–20. PMLR.

Arora, S.; Du, S. S.; Hu, W.; Li, Z.; Salakhutdinov, R.; and Wang, R. 2019. On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904.11955*.

Auer, P.; Chiang, C.-K.; Ortner, R.; and Drugan, M. 2016. Pareto Front Identification from Stochastic Bandit Feedback. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 939–947. PMLR.

Bottou, L.; Peters, J.; Quiñonero-Candela, J.; Charles, D. X.; Chickering, D. M.; Portugaly, E.; Ray, D.; Simard, P.; and Snelson, E. 2013. Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising. *Journal of Machine Learning Research*, 14: 3207–3260.

Brandfonbrener, D.; Whitney, W.; Ranganath, R.; and Bruna, J. 2021. Offline contextual bandits with overparameterized models. In *International Conference on Machine Learning (ICML)*, 1049–1058. PMLR.

Cao, Y.; and Gu, Q. 2019. Generalization Bounds of Stochastic Gradient Descent for Wide and Deep Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32.

Chen, M.; Li, Y.; Wang, E.; Yang, Z.; Wang, Z.; and Zhao, T. 2021. Pessimism meets invariance: Provably efficient offline mean-field multi-agent RL. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34.

Cheng, J.; Xue, B.; Lu, C.; Cui, Z.; and Zhang, Q. 2025. Multi-Objective Neural Bandits with Random Scalarization. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Cheng, J.; Xue, B.; Yi, J.; and Zhang, Q. 2024. Hierarchize Pareto Dominance in Multi-Objective Stochastic Linear Bandits. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, 11489–11497.

Choo, E. U.; and Atkins, D. R. 1983. Proper efficiency in nonconvex multicriteria programming. *Mathematics of Operations Research*, 8(3): 467–470.

Coppolillo, E.; Manco, G.; and Gionis, A. 2024. Relevance meets diversity: A user-centric framework for knowledge exploration through recommendations. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 490–501.

Dai, Z.; Shu, Y.; Low, B. K. H.; and Jaillet, P. 2022. Sample-then-optimize batch neural Thompson sampling. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35.

Drugan, M. M.; and Nowe, A. 2013. Designing multi-objective multi-armed bandits algorithms: A study. In *International Joint Conference on Neural Networks (IJCNN)*. IEEE.

Dudík, M.; Langford, J.; and Li, L. 2011. Doubly robust policy evaluation and learning. In *International Conference on Machine Learning (ICML)*, 1097–1104.

Gottesman, O.; Johansson, F.; Komorowski, M.; Faisal, A.; Sontag, D.; Doshi-Velez, F.; and Celi, L. A. 2019. Guidelines for reinforcement learning in healthcare. *Nature Medicine*, 25(1): 16–18.

Gu, Q.; Karbasi, A.; Khosravi, K.; Mirrokni, V.; and Zhou, D. 2024. Batched neural bandits. *ACM/JMS Journal of Data Science*, 1(1): 1–18.

Hayes, C. F.; Rădulescu, R.; Bargiacchi, E.; Källström, J.; Macfarlane, M.; Reymond, M.; Verstraeten, T.; Zintgraf, L. M.; Dazeley, R.; Heintz, F.; et al. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1): 26.

Ishfaq, H.; Nguyen-Tang, T.; Feng, S.; Arora, R.; Wang, M.; Yin, M.; and Precup, D. 2024. Offline multitask representation learning for reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31.

Jin, Y.; Yang, Z.; and Wang, Z. 2020. Is Pessimism Provably Efficient for Offline RL? *CoRR*, abs/2012.15085.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kong, G.; Xu, D.-L.; and Yang, J.-B. 2008. Clinical decision support systems: a review on knowledge representation and inference under uncertainties. *International Journal of Computational Intelligence Systems*, 1(2): 159–167.

Langford, J.; and Zhang, T. 2007. The Epoch-Greedy Algorithm for Multi-armed Bandits with Side Information. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 20.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit algorithms*. Cambridge University Press.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Liu, X.; Dai, X.; Zuo, J.; Wang, S.; Joe-Wong, C.; Lui, J. C.; and Chen, W. 2025. Offline Learning for Combinatorial Multi-armed Bandits. In *International Conference on Machine Learning (ICML)*.

Lu, S.; Wang, G.; Hu, Y.; and Zhang, L. 2019. Multi-objective generalized linear bandits. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 3080–3086.

Nguyen-Tang, T.; Gupta, S.; Nguyen, A. T.; and Venkatesh, S. 2022. Offline Neural Contextual Bandits: Pessimism, Optimization and Generalization. In *International Conference on Learning Representations (ICLR)*.

Rashidinejad, P.; Zhu, B.; Ma, C.; Jiao, J.; and Russell, S. 2021. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34.

Riquelme, C.; Tucker, G.; and Snoek, J. 2018. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127*.

Sakhi, O.; Alquier, P.; and Chopin, N. 2023. PAC-Bayesian offline contextual bandits with guarantees. In *International Conference on Machine Learning (ICML)*, 29777–29799. PMLR.

Slivkins, A.; et al. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2): 1–286.

Strehl, A. L.; Langford, J.; Li, L.; and Kakade, S. M. 2010. Learning from logged implicit exploration data. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Swaminathan, A.; and Joachims, T. 2015. Batch Learning from Logged Bandit Feedback through Counterfactual Risk Minimization. *Journal of Machine Learning Research*, 16: 1731–1755.

Tekin, C.; and Turğay, E. 2018. Multi-objective contextual multi-armed bandit with a dominant objective. *IEEE Transactions on Signal Processing*, 66(14): 3799–3813.

Turgay, E.; Oner, D.; and Tekin, C. 2018. Multi-objective Contextual Bandit Problem with Similarity Information. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 1673–1681.

Uehara, M.; and Sun, W. 2022. Pessimistic Model-based Offline Reinforcement Learning under Partial Coverage. In *International Conference on Learning Representations (ICLR)*.

Valko, M.; Korda, N.; Munos, R.; Flaounas, I.; and Cristianini, N. 2013. Finite-Time Analysis of Kernelised Contextual Bandits. In *Uncertainty in Artificial Intelligence (UAI)*.

Wang, L.; Krishnamurthy, A.; and Slivkins, A. 2024. Oracle-efficient pessimism: Offline policy optimization in contextual bandits. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR.

Wang, Y.-X.; Agarwal, A.; and Dudık, M. 2017. Optimal and adaptive off-policy evaluation in contextual bandits. In *International Conference on Machine Learning (ICML)*, 3589–3597. PMLR.

Xu, M.; and Klabjan, D. 2023. Pareto regret analyses in multi-objective multi-armed bandit. In *International Conference on Machine Learning (ICML)*, 38499–38517. PMLR.

Xue, B.; Bu, D.; Cheng, J.; Wan, Y.; and Zhang, Q. 2025. Multi-objective Linear Reinforcement Learning with Lexicographic Rewards. In *International Conference on Machine Learning (ICML)*.

Xue, B.; Cheng, J.; Liu, F.; Wang, Y.; and Zhang, Q. 2024. Multiobjective Lipschitz Bandits under Lexicographic Ordering. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Xue, B.; Wang, Y.; Wan, Y.; Yi, J.; and Zhang, L. 2023. Efficient Algorithms for Generalized Linear Bandits with Heavy-tailed Rewards. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yang, R.; Sun, X.; and Narasimhan, K. 2019. A Generalized Algorithm for Multi-Objective Reinforcement Learning and Policy Adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32.

Zhan, X.; Zhu, X.; and Xu, H. 2021. Model-based offline planning with trajectory pruning. *arXiv preprint arXiv:2105.07351*.

Zhou, D.; Li, L.; and Gu, Q. 2020. Neural Contextual Bandits with UCB-based Exploration. In *International Conference on Machine Learning (ICML)*, volume 119, 11492–11502.