# BIAWDRO: CONTROLLING OOD ERROR IN OFFLINE MODEL-BASED OPTIMIZATION VIA WASSERSTEIN DISTRIBUTIONALLY ROBUST OPTIMIZATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Offline model-based optimization (MBO) seeks to discover high-performing designs that maximize a black-box objective function using only pre-collected datasets, with applications such as materials discovery, biological sequence design, and robot morphology optimization. The central challenge in offline MBO is out-of-distribution (OOD) error, where surrogate models yield overconfident predictions that mislead the optimization process toward spurious optima. To mitigate this risk, conservative methods have emerged as a prevalent approach. However, these methods often rely on heuristic techniques. We introduce BiaWDRO, which adapts Wasserstein Distributionally Robust Optimization (DRO) to offline MBO by controlling worst-case predictive bias within Wasserstein ambiguity sets. Applying Lagrangian duality, we derive an objective that mitigates OOD overestimation through tractable adversarial perturbations of training samples. This approach establishes connections between DRO and conservative modeling, providing a principled approach that replaces previous heuristic methods. We further derive an explicit lower bound on empirical fitting error, quantifying the trade-off between worst-case predictive bias and fitting accuracy. Experiments on Design-Bench demonstrate competitive performance across diverse offline MBO tasks.

## 1 INTRODUCTION

Optimizing black-box functions to achieve desired properties is a central task across scientific and industrial domains (Yun et al., 2024), including biological sequence design (Barrera et al., 2016), materials discovery (Hamidieh, 2018), and robot morphology optimization (Liao et al., 2019). While established methods often iteratively query the black-box functions (Angermueller et al., 2020), practical constraints, such as experimental expenses, safety concerns, and time limitations, frequently prohibit such online interaction (Tan et al., 2025; Chen et al., 2022). This gives rise to the setting of offline model-based optimization (MBO), which focuses on searching high-performing designs using only a pre-collected dataset without further access to the ground-truth oracle (Trabucco et al., 2022).

The main challenge of offline MBO lies in out-of-distribution (OOD) errors (Tan et al., 2025), where surrogate models tend to yield overconfident predictions that mislead the optimizer toward spurious optima (Yun et al., 2024; Tan et al., 2025). To mitigate this risk, conservative methods have become a widely adopted approach (Kim et al., 2025). COMs (Trabucco et al., 2021) learns a conservative surrogate by penalizing predictions on adversarial inputs discovered via gradient ascent. BDI (Chen et al., 2022) introduces a backward mapping regularizer that encourages generated designs to stay close to the offline data distribution. PGS (Chemingui et al., 2024) reformulates offline optimization as an offline RL problem, where the action space corresponds to learning rates, and applies conservative Q-learning to prevent overestimation of Q-values on unseen actions. GABO (Yao et al., 2024) introduces adaptive source critic regularization to dynamically constrain the optimization trajectory via a Wasserstein-based critic. Despite their success, these methods lack a principled mechanism to systematically control the trade-off between exploitation and overestimation.

Distributionally Robust Optimization (DRO) (Ben-Tal et al., 2013; Shapiro, 2017) provides a concrete framework for managing model uncertainty by optimizing performance against the worst-case distribution within an ambiguity set. We integrate this framework into offline MBO to establish a



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

principled conservative approach that addresses the overestimation problem. Our central motivation is the explicit control of predictive bias, defined as the increase in the model's expected prediction when the input distribution shifts away from the training distribution. In offline MBO, the critical failure mode is the tendency of surrogate models to yield erroneously high predictions for OOD inputs. Consequently, rather than relying solely on empirical risk minimization, we regularize the training objective by penalizing the worst-case predictive bias—the maximum bias attainable within a specific ambiguity set centered at the training data. This formulation combines data fitting with explicit regularization of predictive bias in the surrounding regions.

To instantiate this framework, we propose BiaWDRO. We employ the Wasserstein distance to define the ambiguity set. Unlike $\phi$-divergence-based approaches (e.g., KL-divergence) which require absolute continuity with respect to the nominal distribution, the Wasserstein distance provides a valid metric between the discrete empirical distribution and continuous perturbed distributions (Mohajerin Esfahani & Kuhn, 2018; Kuhn et al., 2019). This property allows it to naturally accommodate the exploration pattern of offline MBO. Applying Lagrangian duality, we convert the intractable optimization over distributions into a tractable dual objective (Levy et al., 2020). Specifically, we maximize the surrogate's prediction penalized by a transport cost, identifying directions of predictive increase to regularize during training. The resulting nested optimization problem can be solved using alternating stochastic gradient methods. Furthermore, we clarify the structural connection between Conservative Objective Models (COMs) and BiaWDRO through their similar prediction-difference regularization structures. We also establish a quantitative trade-off between conservatism and fitting accuracy.

Our major contributions are summarized as follows:

- We propose a novel principled framework for conservative surrogate learning that adapts DRO to offline MBO. By explicitly controlling the worst-case predictive bias, our approach provides a concrete mechanism to mitigate the OOD overestimation problem.
- We instantiate this framework as BiaWDRO using the Wasserstein ambiguity set. Applying Lagrangian duality, we transform the intractable search over distributions into a tractable dual objective that enforces conservatism through adversarial perturbations penalized by optimal transport costs.
- Theoretically, we clarify the structural connection between BiaWDRO and existing conservative methods (e.g., COMs) and establish a quantitative trade-off between conservatism and fitting accuracy. Empirically, we validate BiaWDRO on Design-Bench, achieving the best mean rank among 22 compared methods across five tasks.

## 2 PRELIMINARIES

### 2.1 PROBLEM FORMULATION

MBO problems are defined on an offline dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathcal{X} \subseteq \mathbb{R}^d$ denotes a design with dimensionality $d$, and $y_i \in \mathbb{R}$ represents its corresponding objective value. The goal is to find a design $x^*$ that maximizes an unknown objective function $f(x)$: $x^* = \arg\max_{x \in \mathcal{X}} f(x)$. Since $f$ is unavailable, a parametric surrogate model $f_\theta$ is learned from $\mathcal{D}$ via empirical risk minimization: $\min_\theta \frac{1}{n} \sum_{i=1}^n \ell(f_\theta(x_i), y_i)$, where $\ell(\cdot, \cdot)$ is a general loss function, and optimization proceeds via gradient ascent on the surrogate: $x_{t+1} = x_t + \eta \nabla_x f_\theta(x_t)$. However, this process drives designs toward OOD regions where the surrogate model lacks training data. In such regions, the surrogate may produce unreliable predictions, leading to suboptimal or invalid designs.

### 2.2 WASSERSTEIN DISTRIBUTIONALLY ROBUST OPTIMIZATION

DRO (Ben-Tal et al., 2013; Shapiro, 2017; Levy et al., 2020) is a framework for training models that are robust to distributional shift. The standard DRO formulation optimizes over an ambiguity set $\mathcal{U}(P_0)$ of distributions centered around the empirical distribution $P_0$: $\min_{\theta \in \Theta} \sup_{P \in \mathcal{U}(P_0)} \mathbb{E}_{(x,y)\sim P}[\ell(f_\theta(x), y)]$, where $\ell(\cdot, \cdot)$ denotes the loss function and $f_\theta$ is the model parameterized by $\theta$. This formulation seeks to find model parameters that minimize the worst-case expected loss over all distributions in the ambiguity set.

Wasserstein DRO (Mohajerin Esfahani & Kuhn, 2018; Blanchet & Murthy, 2019) defines the ambiguity set using the Wasserstein distance, which measures distributional discrepancy through optimal transport. The Wasserstein distance between distributions $P$ and $Q$ is defined as

$$W_c(P, Q) = \inf_{\pi \in \Pi(P,Q)} \left\{ \int_{\mathcal{X}\times\mathcal{X}} c(x, x')\, d\pi(x, x') \right\}, \tag{1}$$

where $\Pi(P, Q)$ denotes the set of joint distributions (couplings) with marginals $P$ and $Q$, i.e., $\pi$ satisfies $\int_{\mathcal{X}} d\pi(x, x') = dP(x)$ and $\int_{\mathcal{X}} d\pi(x, x') = dQ(x')$. The cost function $c(x, x')$ measures the per-unit cost of transporting probability mass from $x$ to $x'$, and the integral $\int_{\mathcal{X}\times\mathcal{X}} c(x, x')\, d\pi(x, x')$ computes the total expected transport cost under plan $\pi$. Thus, $W_c(P, Q)$ represents the minimum cost among all valid transport plans. Defining the ambiguity set as a Wasserstein ball of radius $\rho$:

$$\mathcal{U}_\rho(P_0) = \{P \mid W_c(P, P_0) \le \rho\}, \tag{2}$$

the WDRO problem becomes

$$\min_{\theta\in\Theta} \sup_{P: W_c(P,P_0)\le\rho} \mathbb{E}_{(x,y)\sim P}[\ell(f_\theta(x), y)]. \tag{3}$$

This optimization formulation seeks to find model parameters that minimize the worst-case expected loss over all distributions within Wasserstein distance $\rho$ from the empirical distribution, thereby providing robustness against possible distributional perturbations in the data.

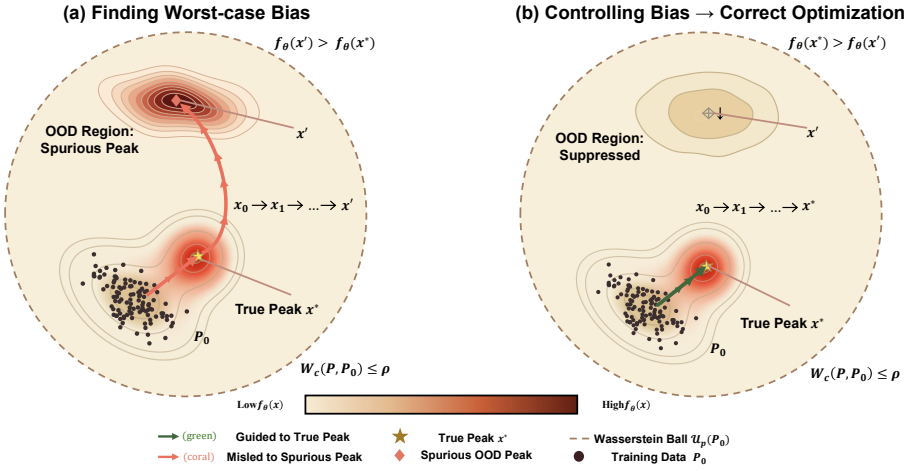



Figure 1: **Overview of BiaWDRO. (a)** BiaWDRO searches for adversarial perturbations using an explicit transport penalty (coral arrow). **(b)** Predictive-bias regularization controls the growth of surrogate predictions during the search for high-performing designs (green arrow).

## 3 BiaWDRO: Bias-controlling via Wasserstein Distributionally Robust Optimization

In this section, we present BiaWDRO, a principled framework that adapts DRO to effectively address the overestimation problem in offline MBO. The core idea is to explicitly regularize the worst-case predictive bias during the surrogate training. Specifically, predictive bias refers to the increase in the surrogate's expected prediction when the input distribution shifts away from the training data, and we consider its worst-case value within an ambiguity set centered at the empirical distribution.

To instantiate this idea, we adopt the Wasserstein distance to define the ambiguity set. This choice is motivated by its ability to provide a valid metric between the discrete empirical distribution and continuous perturbed distributions, which naturally accommodates the exploration pattern in offline MBO. Applying Lagrangian duality, the intractable optimization over distributions reduces to efficient local perturbations around training samples. As illustrated in Figure 1, for each training point, we search for perturbations that maximize prediction penalized by transport cost (Figure 1a). This

worst-case bias penalty regularizes the growth of surrogate predictions in OOD regions during the search for high-performing designs (Figure 1b). Specifically, we first formulate the predictive bias regularization and derive a tractable training objective via Lagrangian duality (Section 3.1), then establish its connection to existing heuristic methods (Section 3.2) and characterize the fitting cost of conservatism (Section 3.3).

### 3.1 Worst-Case Predictive Bias Regularization

We define the worst-case predictive bias as the maximum increase in expected prediction under distribution shift within a Wasserstein ball of radius $\rho$:

$$\Psi_c(\theta) = \sup_{P: W_c(P, P_0) \le \rho} (\mathbb{E}_P[f_\theta] - \mathbb{E}_{P_0}[f_\theta]). \tag{4}$$

Unlike standard WDRO, which finds a worst-case distribution maximizing a loss function, we aim to find the worst-case distribution that maximizes the predictive bias.

We propose to train a surrogate that accurately fits the training data while penalizing large bias.

$$\hat{\theta} = \arg\min_\theta \left( \mathbb{E}_{(x,y)\sim P_0}[(f_\theta(x) - y)^2] + \lambda \cdot \Psi_c(\theta) \right), \tag{5}$$

where $\lambda \ge 0$ controls the regularization strength.

**Dual reformulation.** Direct optimization over distributions in the definition of $\Psi_c(\theta)$ is intractable. We derive a tractable dual form using similar techniques in (Sinha et al., 2017). Introducing Lagrange multiplier $\gamma \ge 0$ for the Wasserstein constraint, strong duality (Slater's condition holds since $W_c(P_0, P_0) = 0 < \rho$) yields

$$\Psi_c(\theta) = \inf_{\gamma \ge 0} \sup_P \left\{ \mathbb{E}_P[f_\theta] - \mathbb{E}_{P_0}[f_\theta] - \gamma W_c(P, P_0) + \gamma\rho \right\}. \tag{6}$$

By the definition of Wasserstein distance,

$$\sup_P \left\{ \mathbb{E}_P[f_\theta] - \gamma W_c(P, P_0) \right\} = \sup_{\pi \in \Pi} \mathbb{E}_{(x,x')\sim\pi}[f_\theta(x') - \gamma c(x', x)], \tag{7}$$

where the supremum is over all couplings $\Pi(P_0, P)$. The optimal coupling has a deterministic marginal: for each $x$, the marginal of $x'$ concentrates mass on the maximizer of $f_\theta - \gamma c$. Formally, $\sup_{\pi\in\Pi} \mathbb{E}_\pi[f_\theta(x') - \gamma c(x', x)] = \mathbb{E}_{x\sim P_0}[\sup_{x'}(f_\theta(x') - \gamma c(x', x))]$. Substituting back, we obtain the dual form

$$\Psi_c(\theta) = \inf_{\gamma \ge 0} \left\{ \gamma\rho + \mathbb{E}_{x\sim P_0}\left[ \sup_{x'} (f_\theta(x') - \gamma c(x', x)) \right] - \mathbb{E}_{P_0}[f_\theta] \right\}. \tag{8}$$

The inner supremum seeks, for each $x$, an adversarial perturbation $x'$ that maximizes prediction while paying a cost $\gamma c(x', x)$.

**Training objective.** Substituting Eq. (8) into Eq. (5), the full optimization problem is

$$\min_\theta \left\{ \mathbb{E}_{(x,y)\sim P_0}[(f_\theta(x) - y)^2] + \lambda \cdot \inf_{\gamma \ge 0} \left[ \gamma\rho + \mathbb{E}_{x\sim P_0}[\sup_{x'}(f_\theta(x') - \gamma c(x', x))] - \mathbb{E}_{P_0}[f_\theta] \right] \right\}. \tag{9}$$

Eq. (9) constitutes a nested min-min-max optimization problem involving the model parameters $\theta$, the dual variable $\gamma$, and the adversarial perturbation $x'$. Due to the complex interdependence of these variables, we employ an alternating stochastic gradient scheme. For fixed $\theta$ and $\gamma$, the inner maximization $\sup_{x'}\{f_\theta(x') - \gamma c(x', x)\}$ is optimized using $K$ gradient-ascent steps, yielding the adversarial perturbation map $\tilde{x} \approx \arg\max_{x'}(f_\theta(x') - \gamma c(x', x))$.

To update the outer variables $\theta$ and $\gamma$, we hold $\tilde{x}$ fixed rather than differentiate through its optimization path. This is motivated by Danskin's Theorem (Danskin, 1966): under its regularity conditions, for $\mathcal{V}(\theta, \gamma) = \sup_{x'}\{f_\theta(x') - \gamma c(x', x)\}$, the derivatives are the partial derivatives of the inner objective evaluated at an exact, unique maximizer $x^*$. Specifically, $\nabla_\theta \mathcal{V}(\theta, \gamma) = \nabla_\theta f_\theta(x')|_{x'=x^*}$ and $\partial\mathcal{V}(\theta, \gamma)/\partial\gamma = -c(x', x)|_{x'=x^*}$.

Building on this, we derive the explicit update rules. Let $\mathcal{B}_t = \{(x_i, y_i)\}_{i=1}^B$ be the mini-batch sampled at step $t$, and let $\tilde{x}_i^{(t)} \approx \arg\max_{x'}(f_{\theta^{(t)}}(x') - \gamma^{(t)} c(x', x_i))$ denote the corresponding

**Algorithm 1** BiaWDRO: Controlling Predictive Bias via Wasserstein DRO

**Input:** Dataset $\mathcal{D}$, Wasserstein radius $\rho$, regularization weight $\lambda$
**Initialize:** Model parameters $\theta$, dual variable $\gamma > 0$
**Hyperparameters:** Learning rates $\eta_\theta$, $\eta_\gamma$, $\eta_{adv}$, adversarial steps $K$, batch size $B$
**for** epoch $= 1$ to $N_{\text{epochs}}$ **do**
  **for** each mini-batch $\{(x_i, y_i)\}_{i=1}^{B} \subset \mathcal{D}$ **do**
    *// Step 1: Approximate inner maximization via gradient ascent*
    **for** $i = 1$ to $B$ **do**
      $\tilde{x}_i \leftarrow x_i$ *// Initialize from training sample*
      **for** $k = 1$ to $K$ **do**
        $\tilde{x}_i \leftarrow \tilde{x}_i + \eta_{adv} \nabla_{x'} \left( f_\theta(x') - \gamma c(x', x_i) \right) \big|_{x'=\tilde{x}_i}$
      **end for**
    **end for**
    *// Step 2: Update dual variable $\gamma$ via projected gradient descent*
    $\gamma \leftarrow \left[ \gamma - \eta_\gamma \left( \rho - \frac{1}{B} \sum_{i=1}^{B} c(\tilde{x}_i, x_i) \right) \right]_+$
    *// Step 3: Update model parameters $\theta$*
    $\mathcal{L} \leftarrow \frac{1}{B} \sum_{i=1}^{B} (f_\theta(x_i) - y_i)^2 + \lambda \left[ \frac{1}{B} \sum_{i=1}^{B} f_\theta(\tilde{x}_i) - \frac{1}{B} \sum_{i=1}^{B} f_\theta(x_i) \right]$
    $\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}$
  **end for**
**end for**
**Output:** Trained surrogate model $f_{\hat{\theta}}$

adversarial perturbations. Since the objective is minimized over $\gamma \geq 0$, we apply projected gradient descent to obtain

$$\gamma^{(t+1)} \leftarrow \left[ \gamma^{(t)} - \eta_\gamma \left( \rho - \frac{1}{B} \sum_{i=1}^{B} c(\tilde{x}i^{(t)}, x_i) \right) \right] +, \tag{10}$$

where $\eta_\gamma$ is the learning rate and $[\cdot]_+ = \max(0, \cdot)$. For $\theta$, we perform gradient descent on the mini-batch surrogate of Eq. (5):

$$\hat{\mathcal{L}}(\theta; \tilde{x}^{(t)}) = \frac{1}{B} \sum_{i=1}^{B} \left[ (f_\theta(x_i) - y_i)^2 + \lambda \left( f_\theta(\tilde{x}i^{(t)}) - f\theta(x_i) \right) \right], \tag{11}$$

where $\tilde{x}_i^{(t)}$ is held fixed and $\eta_\theta$ is the learning rate. Finally, to promote stable outer-loop optimization, we adopt an alternating update scheme inspired by two-timescale stochastic approximation (Borkar, 2008; Borkar & Meyn, 2000). We use a larger learning rate for $\gamma$ than for $\theta$ ($\eta_\gamma \gg \eta_\theta$), aiming to let the dual variable adjust to the current surrogate while the model parameters evolve more slowly. The procedure for a fixed regularization weight $\lambda$ is detailed in Algorithm 1. In our experiments, $\lambda$ is updated according to Eq. (16) in Appendix C.2.

### 3.2 Connection to Conservative Objective Models

Conservative Objective Models (COMs) (Trabucco et al., 2021) is a representative method for training conservative surrogates in MBO. COMs minimizes:

$$\mathbb{E}_{(x,y)\sim P_0}[(f_\theta(x) - y)^2] + \alpha \left( \mathbb{E}_{x\sim\mu}[f_\theta(x)] - \mathbb{E}_{x\sim P_0}[f_\theta(x)] \right), \tag{12}$$

where $\mu$ is an empirical distribution of adversarial samples generated by running gradient ascent for a fixed number of steps, starting from the training points. Comparing this with our objective in Eq. (5), we observe that COMs and BiaWDRO share the same prediction-difference regularization structure, but differ in how adversarial samples are generated.

This connection can be further illustrated by an idealized constrained optimization problem:

$$\min_\theta \left( \mathbb{E}_{(x,y)\sim P_0}[(f_\theta(x) - y)^2] + \alpha \left( \mathbb{E}_{x\sim P_0} \left[ \sup_{x': c(x',x)\leq\epsilon} f_\theta(x') \right] - \mathbb{E}_{P_0}[f_\theta] \right) \right), \tag{13}$$

where $c(x', x) \leq \epsilon$ specifies a local exploration region; choosing $c(x', x) = \|x' - x\|_2$ gives an $\ell_2$ ball. BiaWDRO instead regularizes predictive bias over a Wasserstein ambiguity set, with an explicit transport-cost penalty in its dual objective.

Our formulation offers distinct advantages over COMs: *Metric Flexibility and Explicit Control*. BiaWDRO explicitly incorporates a customizable transportation cost $c(x', x)$. This allows the use of scale-normalized metrics (e.g., MSE), with MSE measuring the average squared displacement per input coordinate. The conservatism level can thus be adjusted through the size of the Wasserstein ambiguity set.

### 3.3 Fitting Cost of Conservatism

The regularizer $\Psi_c$ penalizes increases in surrogate predictions under distribution shift, but excessive conservatism can also impair fitting. Transporting mass between training inputs with different predictions produces a positive gain when moving from the lower prediction to the higher one. Consequently, suppressing $\Psi_c$ restricts variation in training predictions. We characterize this effect for the fixed-$\lambda$ objective in Eq. (5).

Let $R_n(\theta) = n^{-1} \sum_{i=1}^{n} (f_\theta(x_i) - y_i)^2$ denote the empirical squared error and $J_\lambda(\theta) = R_n(\theta) + \lambda \Psi_c(\theta)$ the penalized objective. Assume $n \geq 2$, $\rho > 0$, and a finite, nonnegative, lower-semicontinuous transport cost $c$ with $c(x, x) = 0$. The empirical input distribution is $P_0 = n^{-1} \sum_i \delta_{x_i}$, where $\delta_{x_i}$ denotes a point mass at $x_i$. We assume $\Psi_c(\theta) < \infty$; the remaining transport and integrability conditions are stated in Appendix D.1.

Let $\bar{y} = n^{-1} \sum_i y_i$ be the mean label and $R_{\text{const}} = n^{-1} \sum_i (y_i - \bar{y})^2$ the empirical loss of the best constant predictor. Define

$$a_* = \min_{i \neq j} \min \left\{ \frac{1}{n}, \frac{\rho}{c(x_j, x_i)} \right\} > 0, \qquad \Lambda = \frac{1}{n a_*} \sum_i |y_i - \bar{y}|,$$

with $\rho/0 = +\infty$. Here $a_*$ is a mass that can be transported between any training pair within budget, and $\Lambda$ relates the regularizer to the possible improvement over constant fitting. Write $[a]_+ = \max\{a, 0\}$ for the positive part.

**Theorem 3.1** (Fitting cost and training-prediction collapse)**.** *Under the conditions above,*

$$R_n(\theta) \geq [R_{\text{const}} - \Lambda \Psi_c(\theta)]_+ . \tag{14}$$

*If the model class contains the globally constant predictor $f_\theta \equiv \bar{y}$, then, for $\lambda > \Lambda$, every global minimizer of $J_\lambda$ satisfies $f_\theta(x_i) = \bar{y}$ for every $i$, $\Psi_c(\theta) = 0$, and $R_n(\theta) = R_{\text{const}}$.*

The proof bounds the range of training predictions by $\Psi_c(\theta)/a_*$. This limits the improvement over constant fitting to $\Lambda \Psi_c(\theta)$, which is outweighed by the penalty when $\lambda > \Lambda$. The result explains the fitting cost of excessive conservatism and motivates balancing the fitting loss with the predictive-bias penalty. Appendix D gives the complete proof and extensions to general convex losses.

## 4 Related Work

Offline MBO methods address OOD overestimation through conservative surrogate learning, generative and inverse modeling, gradient regularization, ranking, and trajectory-based optimization. Representative conservative approaches include COMs (Trabucco et al., 2021), BDI (Chen et al., 2022), and GABO (Yao et al., 2024). Our approach connects conservative surrogate learning to Wasserstein DRO, whose ambiguity sets support distributional shifts beyond the empirical support (Mohajerin Esfahani & Kuhn, 2018; Sinha et al., 2017). Appendix E provides the complete review of offline MBO and DRO literature.

## 5 Experiments

In this section, we empirically evaluate BiaWDRO on the Design-Bench benchmark (Trabucco et al., 2022). Our experiments aim to demonstrate the effectiveness of BiaWDRO against state-of-the-art methods, investigate the impact of key design choices including the cost function and Wasserstein radius $\rho$, and verify that the Wasserstein constraint successfully prevents excessive OOD exploration. We first describe the experimental setup, including tasks, evaluation protocol, and baselines, then present comparison results and ablation studies, and finally analyze the exploration behavior of optimized designs. Additional experiments examining performance under varying evaluation budgets are provided in Appendix C.3.

Table 1: **Experiments on Design-Bench Tasks.** We report the maximum normalized score ($100^{th}$ percentile) among $Q$=128 candidates as mean $\pm$ standard error (SE) over 8 independent runs. Within each task, methods are ranked by descending displayed mean score, with ties assigned the smallest shared rank. Mean Rank averages these ranks across the five tasks. **Blue** denotes the best entry in the column, and **Violet** denotes the second best.

| **Method** | TFBind8 | TFBind10 | Superconductor | Ant | D'Kitty | Mean Rank |
|---|---|---|---|---|---|---|
| $\mathcal{D}$ (best) | 0.439 | 0.467 | 0.399 | 0.565 | 0.884 | - |
| BO-qEI | $0.794 \pm 0.103$ | $0.631 \pm 0.041$ | $0.486 \pm 0.025$ | $0.812 \pm 0.000$ | $0.896 \pm 0.000$ | 17.0 / 22 |
| CMA-ES | $0.919 \pm 0.055$ | $0.649 \pm 0.020$ | $0.478 \pm 0.010$ | **$2.222 \pm 1.550$** | $0.724 \pm 0.001$ | 13.2 / 22 |
| REINFORCE | $0.947 \pm 0.029$ | $0.628 \pm 0.025$ | $0.485 \pm 0.011$ | $0.247 \pm 0.031$ | $0.558 \pm 0.193$ | 17.2 / 22 |
| Grad Ascent | **$0.983 \pm 0.015$** | $0.648 \pm 0.044$ | $0.509 \pm 0.018$ | $0.295 \pm 0.021$ | $0.877 \pm 0.023$ | 11.4 / 22 |
| CbAS | $0.895 \pm 0.043$ | $0.638 \pm 0.040$ | $0.468 \pm 0.058$ | $0.825 \pm 0.030$ | $0.898 \pm 0.011$ | 16.6 / 22 |
| MINs | $0.884 \pm 0.039$ | $0.660 \pm 0.048$ | $0.500 \pm 0.036$ | $0.908 \pm 0.031$ | $0.942 \pm 0.005$ | 13.2 / 22 |
| DDOM | $0.966 \pm 0.015$ | $0.666 \pm 0.024$ | $0.476 \pm 0.029$ | $0.926 \pm 0.027$ | $0.948 \pm 0.011$ | 10.6 / 22 |
| BONET | $0.831 \pm 0.109$ | $0.606 \pm 0.044$ | $0.405 \pm 0.017$ | **$0.957 \pm 0.004$** | $0.950 \pm 0.014$ | 14.8 / 22 |
| GTG | $0.923 \pm 0.036$ | $0.620 \pm 0.054$ | $0.467 \pm 0.059$ | $0.876 \pm 0.045$ | $0.939 \pm 0.012$ | 16.4 / 22 |
| ROOT | $0.982 \pm 0.009$ | $0.696 \pm 0.068$ | $0.466 \pm 0.030$ | **$0.957 \pm 0.007$** | **$0.972 \pm 0.004$** | 6.0 / 22 |
| COMs | $0.941 \pm 0.048$ | $0.647 \pm 0.028$ | $0.465 \pm 0.114$ | $0.933 \pm 0.030$ | $0.946 \pm 0.007$ | 13.4 / 22 |
| NEMO | $0.941 \pm 0.000$ | $0.705 \pm 0.000$ | $0.502 \pm 0.002$ | $0.952 \pm 0.002$ | $0.950 \pm 0.001$ | 7.6 / 22 |
| RoMA | $0.924 \pm 0.040$ | $0.666 \pm 0.035$ | $0.510 \pm 0.015$ | $0.917 \pm 0.030$ | $0.927 \pm 0.013$ | 10.8 / 22 |
| BDI | $0.973 \pm 0.000$ | $0.630 \pm 0.025$ | $0.508 \pm 0.011$ | $0.932 \pm 0.000$ | $0.939 \pm 0.000$ | 10.2 / 22 |
| ICT | $0.944 \pm 0.015$ | $0.598 \pm 0.020$ | $0.507 \pm 0.014$ | $0.946 \pm 0.021$ | $0.960 \pm 0.014$ | 10.2 / 22 |
| PGS | $0.975 \pm 0.023$ | $0.681 \pm 0.038$ | $0.458 \pm 0.026$ | $0.733 \pm 0.020$ | $0.957 \pm 0.017$ | 11.2 / 22 |
| Match-OPT | $0.967 \pm 0.029$ | $0.656 \pm 0.052$ | $0.505 \pm 0.007$ | $0.942 \pm 0.028$ | $0.958 \pm 0.015$ | 7.6 / 22 |
| RaM-ListNet | $0.967 \pm 0.016$ | $0.697 \pm 0.132$ | **$0.514 \pm 0.025$** | $0.921 \pm 0.038$ | **$0.962 \pm 0.014$** | **5.4 / 22** |
| RaM-RankCosine | $0.973 \pm 0.027$ | $0.651 \pm 0.031$ | $0.508 \pm 0.028$ | $0.910 \pm 0.021$ | $0.958 \pm 0.019$ | 8.0 / 22 |
| GABO | $0.981 \pm 0.012$ | **$0.723 \pm 0.048$** | $0.412 \pm 0.005$ | $0.108 \pm 0.007$ | $0.724 \pm 0.002$ | 13.6 / 22 |
| DynAMO | $0.965 \pm 0.009$ | $0.701 \pm 0.126$ | $0.500 \pm 0.047$ | $0.139 \pm 0.037$ | $0.847 \pm 0.016$ | 13.0 / 22 |
| **BiaWDRO (Ours)** | **$0.987 \pm 0.007$** | **$0.713 \pm 0.061$** | **$0.513 \pm 0.013$** | $0.952 \pm 0.012$ | $0.956 \pm 0.007$ | **3.2 / 22** |

### 5.1 Tasks and Evaluation

**Tasks.** In this work, we evaluate the proposed method on a diverse set of tasks from the Design-Bench (Trabucco et al., 2022) offline MBO benchmark, encompassing both continuous and discrete design problems. The continuous tasks include: (1) Ant Morphology (Brockman et al., 2016), which involves optimizing a 60-dimensional parameter vector that defines the physical structure of a simulated ant robot to maximize its locomotion speed; (2) D'Kitty Morphology (Ahn et al., 2020), a similar robotic morphology optimization task aimed at improving the walking performance of a D'Kitty robot through a 56-dimensional continuous parameter search; and (3) Superconductor (Hamidieh, 2018), where the objective is to identify a superconducting material, represented by an 86-dimensional continuous vector, that achieves a high critical temperature. For discrete design tasks, we consider: (4) TF-Bind-8 and (5) TF-Bind-10 (Barrera et al., 2016), both of which involve discovering DNA sequences of lengths 8 and 10, respectively, that maximize binding affinity to a specified transcription factor. All tasks are conducted using pre-existing offline datasets that provide mappings from design parameters (or sequences) to their corresponding performance scores. Detailed task specifications are provided in Appendix A.

**Evaluation.** We follow the standard evaluation protocol established in prior offline MBO literature (Yun et al., 2024; Tan et al., 2025). Specifically, for each method, we query the learned model to obtain the top $k = 128$ most promising candidate designs. The performance of each method is then quantified by the normalized ground-truth score of the best candidate within this set (i.e., the 100th percentile score). Normalization is performed as follows: $y_{\text{norm}} = \frac{y - y_{\min}}{y_{\max} - y_{\min}}$, where $y_{\min}$ and $y_{\max}$ denote the minimum and maximum scores, respectively, observed in the complete held-out dataset. We report mean normalized scores $\pm$ standard error (SE) over 8 independent random seeds. Details of reproduced and previously reported baseline results are provided in Appendix B.2.

### 5.2 Results and Analysis

**Performance Comparison on Design-Bench Tasks.** Table 1 compares all 22 methods on the five Design-Bench tasks. BiaWDRO achieves the best mean rank of 3.2, compared with 5.4 for

Table 2: **Ablation study on cost functions.** We compare different choices of the cost function $c(x', x)$ in our method. Normalized scores (mean $\pm$ SE) over 8 independent runs are reported. **Blue**: best; **Violet**: second best.

| Cost Function | TFBind8 | TFBind10 | Superconductor | Ant | D'Kitty | Mean Rank |
|---|---|---|---|---|---|---|
| MSE | **0.987 $\pm$ 0.007** | **0.713 $\pm$ 0.061** | **0.513 $\pm$ 0.013** | **0.952 $\pm$ 0.012** | **0.956 $\pm$ 0.007** | **1.4 / 4** |
| Smooth $\ell_1$ (sum) | **0.991 $\pm$ 0.000** | 0.658 $\pm$ 0.048 | 0.508 $\pm$ 0.009 | **0.951 $\pm$ 0.007** | 0.943 $\pm$ 0.007 | 2.8 / 4 |
| Smooth $\ell_1$ (mean) | **0.987 $\pm$ 0.010** | **0.684 $\pm$ 0.055** | **0.509 $\pm$ 0.012** | 0.945 $\pm$ 0.009 | **0.956 $\pm$ 0.009** | **2.2 / 4** |
| $\ell_2$ | **0.991 $\pm$ 0.000** | 0.664 $\pm$ 0.038 | 0.507 $\pm$ 0.010 | 0.911 $\pm$ 0.030 | **0.947 $\pm$ 0.004** | 3.0 / 4 |

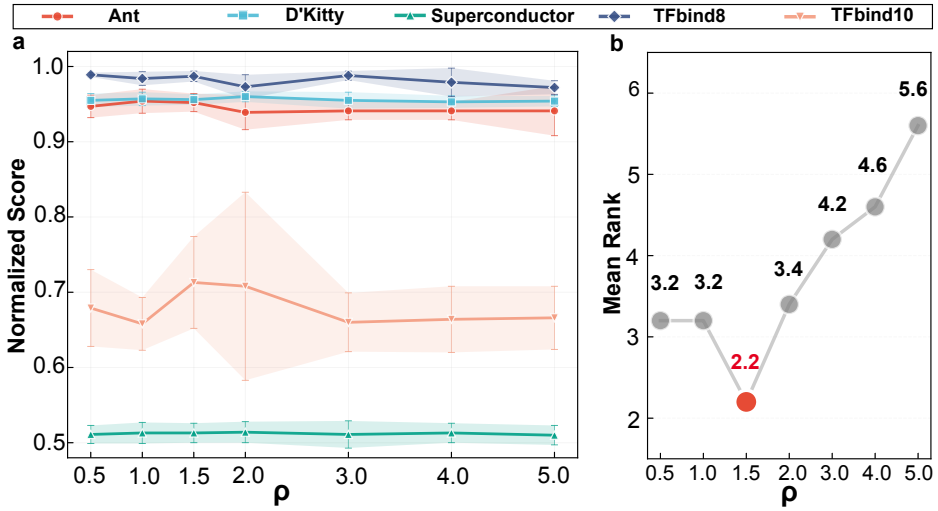



Figure 2: **Effect of Wasserstein radius $\rho$ on Design-Bench tasks.** (a) Normalized scores (mean $\pm$ SE) for five tasks across $\rho \in \{0.5, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0\}$. (b) Mean rank across five tasks as a function of $\rho$ (lower is better).

RaM-ListNet and 6.0 for ROOT. It obtains the highest mean score on TFBind8 and the second-highest on TFBind10 and Superconductor, demonstrating competitive performance across the evaluated tasks. Appendix C.5 examines the sensitivity of this ranking to task inclusion and tie handling (Table 6).

**Choice of Cost Function.** We investigate how different choices of the cost function $c(x', x)$ in Eq. (9) affect optimization performance. We consider four cost functions: MSE, Smooth $\ell_1$ (sum), Smooth $\ell_1$ (mean), and $\ell_2$. Detailed definitions are provided in Appendix B.3.

Table 2 reports results on five Design-Bench tasks. Among the four cost functions, MSE achieves the best mean rank of 1.4, followed by Smooth $\ell_1$ (mean) with 2.2. MSE and Smooth $\ell_1$ (mean) achieve better mean ranks than $\ell_2$ and Smooth $\ell_1$ (sum), respectively. For MSE, the transport budget measures the average squared change per input coordinate, providing an explicit normalization by the input dimension. Based on these findings, we adopt MSE as the default cost function.

**The Influence of $\rho$.** The Wasserstein radius $\rho$ controls the size of the ambiguity set. We standardize input features $x$ and adopt a dimension-normalized cost $c(x', x) = \frac{1}{d}\|x' - x\|_2^2$, ensuring comparable semantics across tasks. Figure 2a reports normalized scores with standard errors. We observe that the performance remains relatively stable across different values of $\rho$, indicating that our method is robust to the precise choice of $\rho$ and does not require extensive tuning. However, Figure 2b, which shows mean rank (lower is better), reveals a clear trend: performance degrades monotonically as $\rho$ increases beyond 1.5. Smaller values ($\rho \in \{0.5, 1.0, 1.5\}$) obtain lower mean ranks than the tested larger values, with $\rho = 1.5$ achieving the best mean rank of 2.2. This pattern is consistent with increased conservatism at larger radii. Overall, these results validate that our method is robust to moderate variations in $\rho$, reducing the burden of hyperparameter selection, while also demonstrating that explicit control over the ambiguity set radius is essential, as overly large radii degrade performance. We set $\rho = 1.5$ as default. Detailed results are in Table 5.

**Exploration Distance Analysis.** As discussed in Section 3.2, COMs and BiaWDRO share the same prediction-difference regularization structure, but differ in how adversarial samples are generated. To understand the behavioral difference between methods, we compare how far COMs and BiaWDRO move designs from their starting points during optimization. Figure 3 presents paired comparisons across all 128 candidates, the same set as in Table 1. For each candidate, we compute the distance from its initial design to the COMs-optimized result and to the BiaWDRO-optimized result, then calculate their difference. We use Hamming distance for discrete tasks (TFBind8/10) and Euclidean distance for continuous tasks (Superconductor, Ant, D'Kitty).

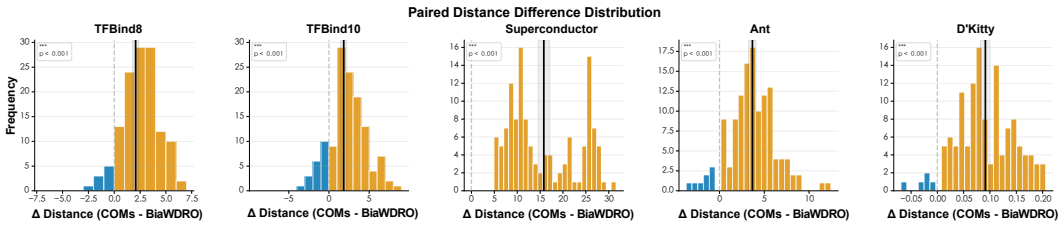



Figure 3: **Paired distance difference distribution.** Distribution of $\Delta$Distance (COMs $-$ BiaWDRO) from initial designs across five tasks. The black solid line marks the mean difference; the gray dashed line marks zero. Positive values (orange) indicate COMs moved farther from initial designs, while negative values (blue) indicate BiaWDRO moved farther. The predominantly positive values demonstrate BiaWDRO's more constrained exploration behavior.

The histograms reveal a consistent pattern across all five tasks: the distribution of differences is predominantly shifted toward positive values (orange bars), indicating that COMs systematically explores farther from initial designs than BiaWDRO. The black solid lines, representing mean differences, are located in the positive region for all tasks. Although some individual samples show negative differences (blue bars), indicating cases where BiaWDRO moved farther, these are substantially outnumbered by positive cases across all tasks.

This analysis demonstrates that BiaWDRO generally keeps optimized designs closer to their initial points than COMs. Combined with the results in Table 1, BiaWDRO achieves higher mean scores than COMs while exploring less aggressively, suggesting that transport-regularized conservative training can improve the effectiveness of subsequent exploration. Additional analyses are provided in Appendix C.4.

## 6 Conclusion

In this work, we proposed BiaWDRO, a principled framework for conservative surrogate learning in offline MBO that addresses the critical challenge of OOD overestimation. By integrating DRO into the offline MBO setting, our approach explicitly controls the worst-case predictive bias within a Wasserstein ambiguity set centered at the training distribution. Through Lagrangian duality, we transformed the intractable optimization over distributions into a tractable dual objective that induces conservatism via adversarial perturbations penalized by optimal transport costs. Our analysis establishes a theoretical trade-off between conservatism and fitting accuracy. Empirical evaluations on five Design-Bench tasks show competitive performance, with the best mean rank among 22 methods. Ablation studies and exploration distance analysis characterize the effects of the transportation cost and Wasserstein radius $\rho$, and show less aggressive exploration compared to COMs. We hope this work can inspire further exploration of distributionally robust methods in related offline optimization problems.

## 7 Limitations

We discuss one main limitation of BiaWDRO that points to a direction for future work. Although BiaWDRO is robust to moderate variations in the Wasserstein radius $\rho$ (Figure 2), $\rho$ is currently a fixed global hyperparameter independent of local data density or surrogate uncertainty; an adaptive $\rho$ guided by uncertainty or density estimates is a promising direction.

### AI USE STATEMENT

During manuscript preparation, we used generative AI tools for language editing, including correcting grammatical errors, refining sentence structure, and improving the clarity and readability of the writing. We reviewed and revised the AI-assisted edits to ensure that they accurately preserved the intended technical meaning. We take full responsibility for the accuracy, originality, and integrity of the final manuscript.

### ETHICS STATEMENT

This work focuses on the methodological and theoretical study of offline model-based optimization. All experiments are computational evaluations using existing benchmark datasets and oracles, with no human-subject studies, animal experiments, or physical deployment. No new personal or sensitive data are collected. We do not identify specific ethical concerns associated with the research presented in this paper.

### REPRODUCIBILITY STATEMENT

We provide details of the proposed method, experimental settings, and theoretical analysis to support reproducibility. Section 3.1 and Algorithm 1 describe the training objective and optimization procedure, while Appendix C.2 specifies the adaptive update of the regularization weight. Appendix A documents the benchmark tasks, and Table 4 reports the training hyperparameters. Appendix B.2 describes the baseline implementations and sources of reported results. Appendix B.3 defines the transport costs and the continuous representation used for discrete tasks. Additional hyperparameter sensitivity and evaluation-budget analyses are provided in Table 5 and Appendices C.2 and C.3. For the theoretical results, Appendix D states the assumptions and provides complete proofs of Theorem 3.1 and its extensions.

## REFERENCES

Michael Ahn, Henry Zhu, Kristian Hartikainen, Hugo Ponte, Abhishek Gupta, Sergey Levine, and Vikash Kumar. Robel: Robotics benchmarks for learning with low-cost robots. In *Conference on robot learning*, pp. 1300–1313. PMLR, 2020.

Christof Angermueller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy Colwell. Model-based reinforcement learning for biological sequence design. In *International conference on learning representations*, 2020.

Luis A Barrera, Anastasia Vedenko, Jesse V Kurland, Julia M Rogers, Stephen S Gisselbrecht, Elizabeth J Rossin, Jaie Woodard, Luca Mariani, Kian Hong Kock, Sachi Inukai, et al. Survey of variation in human transcription factors reveals prevalent dna binding changes. *Science*, 351 (6280):1450–1454, 2016.

Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600, 2019.

Vivek S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*, volume 48 of *Texts and Readings in Mathematics*. Hindustan Book Agency, 2008. doi: 10.1007/978-93-86279-38-5.

Vivek S Borkar and Sean P Meyn. The ode method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 38(2):447–469, 2000.

Stephen Boyd, John Duchi, Mert Pilanci, and Lieven Vandenberghe. Subgradients. Notes for EE364b, Stanford University, April 2022. April 13, 2022.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

David Brookes, Hahnbeom Park, and Jennifer Listgarten. Conditioning by adaptive sampling for robust design. In *International conference on machine learning*, pp. 773–782. PMLR, 2019.

Yassine Chemingui, Aryan Deshwal, Trong Nghia Hoang, and Janardhan Rao Doppa. Offline model-based optimization via policy-guided gradient search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11230–11239, 2024.

Can Chen, Yingxue Zhang, Jie Fu, Xue Steve Liu, and Mark Coates. Bidirectional learning for offline infinite-width model-based optimization. *Advances in Neural Information Processing Systems*, 35: 29454–29467, 2022.

Can Sam Chen, Christopher Beckham, Zixuan Liu, Xue Steve Liu, and Chris Pal. Parallel-mentoring for offline model-based optimization. *Advances in Neural Information Processing Systems*, 36: 76619–76636, 2023.

Can Sam Chen, Christopher Beckham, Zixuan Liu, Xue Liu, and Christopher Pal. Robust guided diffusion for offline black-box optimization. *arXiv preprint arXiv:2410.00983*, 2024.

John M Danskin. The theory of max-min, with applications. *SIAM Journal on Applied Mathematics*, 14(4):641–664, 1966.

Cuong Dao, Phi Le Nguyen, Truong Thao Nguyen, and Nghia Hoang. Incorporating surrogate gradient norm to improve offline optimization techniques. *Advances in Neural Information Processing Systems*, 37:8014–8046, 2024.

Manh Cuong Dao, The Hung Tran, Phi Le Nguyen, Thao Nguyen Truong, and Trong Nghia Hoang. Root: Rethinking offline optimization as distributional translation via probabilistic bridge. *arXiv preprint arXiv:2509.16300*, 2025.

Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

Justin Fu and Sergey Levine. Offline model-based optimization via normalized maximum likelihood estimation. *arXiv preprint arXiv:2102.07970*, 2021.

Rui Gao, Xi Chen, and Anton J Kleywegt. Wasserstein distributionally robust optimization and variation regularization. *Operations Research*, 72(3):1177–1191, 2024.

Roman Garnett. *Bayesian optimization*. Cambridge University Press, 2023.

Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

Kam Hamidieh. A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science*, 154:346–354, 2018.

Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Nam Ho-Nguyen and Stephen J Wright. Adversarial classification via distributional robustness with wasserstein ambiguity. *Mathematical Programming*, 198(2):1411–1447, 2023.

Minh Hoang, Azza Fadhel, Aryan Deshwal, Janardhan Rao Doppa, and Trong Nghia Hoang. Learning surrogates for offline black-box optimization via gradient matching. *arXiv preprint arXiv:2503.01883*, 2025.

Minsu Kim, Jiayao Gu, Ye Yuan, Taeyoung Yun, Zixuan Liu, Yoshua Bengio, and Can Chen. Offline model-based optimization: Comprehensive review. *arXiv preprint arXiv:2503.17286*, 2025.

Siddarth Krishnamoorthy, Satvik Mehul Mashkaria, and Aditya Grover. Diffusion models for black-box optimization. In *International Conference on Machine Learning*, pp. 17842–17857. PMLR, 2023.

Daniel Kuhn, Peyman Mohajerin Esfahani, Viet Anh Nguyen, and Soroosh Shafieezadeh-Abadeh. Wasserstein distributionally robust optimization: Theory and applications in machine learning. In *Operations research & management science in the age of analytics*, pp. 130–166. Informs, 2019.

Aviral Kumar and Sergey Levine. Model inversion networks for model-based optimization. *Advances in neural information processing systems*, 33:5126–5137, 2020.

Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. *Advances in neural information processing systems*, 33:8847–8860, 2020.

Thomas Liao, Grant Wang, Brian Yang, Rene Lee, Kristofer Pister, Sergey Levine, and Roberto Calandra. Data-efficient learning of morphology and controller for a microrobot. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2488–2494. IEEE, 2019.

Satvik Mehul Mashkaria, Siddarth Krishnamoorthy, and Aditya Grover. Generative pretraining for black-box optimization. In *International Conference on Machine Learning*, pp. 24173–24197. PMLR, 2023.

Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, 2018.

Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. *Advances in neural information processing systems*, 29, 2016.

Herbert Scarf. A min-max solution of an inventory problem. In Kenneth J. Arrow, Samuel Karlin, and Herbert Scarf (eds.), *Studies in the Mathematical Theory of Inventory and Production*, pp. 201–209. Stanford University Press, Stanford, CA, 1958.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2015.

Alexander Shapiro. Distributionally robust stochastic programming. *SIAM Journal on Optimization*, 27(4):2258–2275, 2017.

Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Rong-Xi Tan, Ke Xue, Shen-Huan Lyu, Haopu Shang, yaowang, Yaoyuan Wang, Fu Sheng, and Chao Qian. Offline model-based optimization by learning to rank. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sb1HgVDLjN.

Brandon Trabucco, Aviral Kumar, Xinyang Geng, and Sergey Levine. Conservative objective models for effective offline model-based optimization. In *International Conference on Machine Learning*, pp. 10358–10368. PMLR, 2021.

Brandon Trabucco, Xinyang Geng, Aviral Kumar, and Sergey Levine. Design-bench: Benchmarks for data-driven offline model-based optimization. In *International Conference on Machine Learning*, pp. 21658–21676. PMLR, 2022.

Wolfram Wiesemann, Daniel Kuhn, and Melvyn Sim. Distributionally robust convex optimization. *Operations research*, 62(6):1358–1376, 2014.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Michael S Yao, Yimeng Zeng, Hamsa Bastani, Jacob Gardner, James C Gee, and Osbert Bastani. Generative adversarial model-based optimization via source critic regularization. *Advances in neural information processing systems*, 37:44009–44039, 2024.

Michael S Yao, James C Gee, and Osbert Bastani. Diversity by design: Leveraging distribution matching for offline model-based optimization. *arXiv preprint arXiv:2501.18768*, 2025.

Sihyun Yu, Sungsoo Ahn, Le Song, and Jinwoo Shin. Roma: Robust model adaptation for offline model-based optimization. *Advances in Neural Information Processing Systems*, 34:4619–4631, 2021.

Ye Yuan, Can Sam Chen, Zixuan Liu, Willie Neiswanger, and Xue Steve Liu. Importance-aware co-teaching for offline model-based optimization. *Advances in Neural Information Processing Systems*, 36:55718–55733, 2023.

Taeyoung Yun, Sujin Yun, Jaewoo Lee, and Jinkyoo Park. Guided trajectory generation with diffusion models for offline model-based optimization. *Advances in Neural Information Processing Systems*, 37:83847–83876, 2024.

# A Task Details

## A.1 Design-Bench Tasks

For the evaluation of MBO algorithms, we adopt Design-Bench (Trabucco et al., 2022), which serves as the predominant benchmark in this domain. Our experimental configuration aligns with that established by prior work (Yun et al., 2024), encompassing both continuous and discrete optimization scenarios. Specifically, the continuous optimization tasks comprise Ant, D'Kitty, and Superconductor, whereas the discrete optimization tasks consist of TFBind8 and TFBind10. Comprehensive specifications for each task are delineated in Table 3.

Table 3: Detailed Specifications of Benchmark Tasks

| Task | Dataset Size | Dimensions | Type | Oracle | Max |
|---|---|---|---|---|---|
| Superconductor | 17,014 | 86 | Continuous | Random Forest | 185.0 |
| Ant | 10,004 | 60 | Continuous | Exact | 590.0 |
| D'Kitty | 10,004 | 56 | Continuous | Exact | 340.0 |
| TFBind8 | 32,898 | 8 | Discrete | Exact | 1.0 |
| TFBind10 | 50,000 | 10 | Discrete | Exact | 2.128 |

## A.2 Excluded Design-Bench Tasks

Following established protocols in prior research (Yun et al., 2024; Tan et al., 2025), we exclude three benchmarks from the Design-Bench suite for our experimental evaluation: Hopper (Brockman et al., 2016), ChEMBL (Gaulton et al., 2012), and the synthetic Neural Architecture Search (NAS) tasks on CIFAR10 (Hinton et al., 2012). The exclusion of Hopper follows from a documented implementation defect in Design-Bench (see `https://github.com/brandontrabucco/design-bench/issues/8#issuecomment-1086758113` for details), which causes oracle instability characterized by a strong distributional bias toward lower function values and unreliable behavior across evaluations. ChEMBL is omitted due to performance saturation—as demonstrated in previous studies (Mashkaria et al., 2023; Krishnamoorthy et al., 2023), different methods yield indistinguishable results on this task, making it unsuitable for comparative analysis and method discrimination. The NAS benchmark is excluded owing to its prohibitive computational requirements when conducting rigorous multi-seed evaluations, which exceeds our available computational budget.

# B Implementation Details

## B.1 BiaWDRO

We implement the proxy function using a multi-layer perceptron (MLP) consisting of 2 hidden layers, each with 2048 hidden units and ReLU activation functions.

For BiaWDRO training, we specify several key hyperparameters, including the number of training epochs, the model learning rate $\eta_\theta$, the dual variable learning rate $\eta_\gamma$, the number of gradient ascent steps for generating adversarial samples, and the perturbation constraint $\rho$, which controls the allowed average perturbation from the training data. Throughout all experiments, we fix $\eta_\gamma = 0.01$ across all tasks. Following COMs (Trabucco et al., 2021), we set $\tau = 0.5$ for all continuous tasks and $\tau = 2.0$ for all discrete tasks to automatically determine the value of $\lambda$. The detailed hyperparameter configurations for each benchmark are summarized in Table 4.

## B.2 Baselines

We reproduce the following recent baselines using their official open-source implementations with the original hyperparameter settings: PGS (Chemingui et al., 2024), GTG (Yun et al., 2024), GABO (Yao et al., 2024), DynAMO (Yao et al., 2025), ROOT (Dao et al., 2025), MatchOpt (Hoang et al., 2025), and RAM (Tan et al., 2025). Additionally, as a representative method for conservative optimization in offline MBO, we reproduce COMs (Trabucco et al., 2021) using its official implementation following

Table 4: Hyperparameters for BiaWDRO Training.

| Task | Gradient Ascent Steps | Epochs | $\rho$ | $\eta_\theta$ |
|---|---|---|---|---|
| Superconductor | 5 | 5 | 1.5 | $7 \times 10^{-5}$ |
| Ant | 5 | 5 | 1.5 | $7 \times 10^{-5}$ |
| D'Kitty | 5 | 5 | 1.5 | $7 \times 10^{-5}$ |
| TFBind8 | 20 | 20 | 1.5 | $3 \times 10^{-5}$ |
| TFBind10 | 20 | 30 | 1.5 | $3 \times 10^{-5}$ |

the original hyperparameters. For earlier baselines, including BO-qEI (Garnett, 2023; Shahriari et al., 2015), CMA-ES (Hansen, 2016), REINFORCE (Williams, 1992), Gradient Ascent, CbAS (Brookes et al., 2019), MINs (Kumar & Levine, 2020), NEMO (Fu & Levine, 2021), BDI (Chen et al., 2022), ICT (Yuan et al., 2023), DDOM (Krishnamoorthy et al., 2023), and BONET (Mashkaria et al., 2023), we directly adopt the results from (Yun et al., 2024). All experiments follow the same settings as (Yun et al., 2024), with an evaluation budget of Q=128 and 8 independent random seeds. Our reproduced GTG scores are lower than those reported in the original paper (Yun et al., 2024); a similar discrepancy is also observed in the independent evaluation reported in ROOT (Dao et al., 2025, Table 1).

### B.3 Cost Function Definitions

We consider the following cost functions $c(x', x)$ to measure the distance between the optimized design $x'$ and the original design $x \in \mathbb{R}^d$. Let $\delta = x' - x$ denote the element-wise difference.

**MSE** computes the mean of squared differences across all elements: $c(x', x) = \frac{1}{d} \sum_{i=1}^{d} \delta_i^2$.

We also consider two cost functions based on the Charbonnier penalty, which provides a smooth approximation to the $\ell_1$ norm. **Smooth $\ell_1$ (sum)** applies this penalty element-wise and sums the results: $c(x', x) = \sum_{i=1}^{d} \sqrt{\delta_i^2 + \epsilon^2}$, where $\epsilon = 10^{-6}$ is a small constant for numerical stability. **Smooth $\ell_1$ (mean)** is its averaged version: $c(x', x) = \frac{1}{d} \sum_{i=1}^{d} \sqrt{\delta_i^2 + \epsilon^2}$. These implemented costs include a constant offset $\kappa$, where $\kappa = d\epsilon$ for Smooth $\ell_1$ (sum) and $\kappa = \epsilon$ for Smooth $\ell_1$ (mean). Define the centered cost $\bar{c}(x', x) = c(x', x) - \kappa$, which satisfies $\bar{c}(x, x) = 0$. Since transport plans have unit total mass, $W_c(P, P_0) = W_{\bar{c}}(P, P_0) + \kappa$. Consequently, for $\rho > \kappa$, the constraint $W_c(P, P_0) \leq \rho$ is equivalent to $W_{\bar{c}}(P, P_0) \leq \bar{\rho}$, where $\bar{\rho} = \rho - \kappa$. When relating these two ablations to the zero-diagonal transport convention in the theoretical analysis, we use this equivalent centered representation.

Additionally, we include the standard Euclidean transport cost, denoted by $\ell_2$: $c(x', x) = \|x' - x\|_2 = \sqrt{\sum_{i=1}^{d} \delta_i^2}$.

For discrete tasks (TFBind8/10), all optimization is performed in a continuous logit space $\mathbf{x} \in \mathbb{R}^{L \times V}$ following standard practice (Trabucco et al., 2021), and the above cost functions are computed on these logit vectors; the discrete sequence is recovered via $\arg\max$ only at evaluation.

## C Extended Additional Analysis

### C.1 Details of the Influence of $\rho$

Table 5 presents the complete results of the sensitivity analysis for the Wasserstein radius $\rho$. We evaluate $\rho \in \{0.5, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0\}$ on five Design-Bench tasks, reporting normalized scores (mean $\pm$ SE) over 8 independent runs. For each $\rho$, we compute its rank within each task and report the mean rank across all tasks. The best overall performance is achieved at $\rho = 1.5$ with a mean rank of 2.2, followed by $\rho \in \{0.5, 1.0\}$ at 3.2. Performance degrades monotonically for $\rho > 1.5$, with $\rho = 5.0$ yielding the worst mean rank of 5.6.

### C.2 Sensitivity Analysis of $\tau$

In Eq. (5), the regularization weight $\lambda$ directly controls the trade-off between data fitting and conservatism. However, its optimal value depends on the magnitude of the surrogate predictions, which vary across tasks, making per-task tuning necessary. To avoid this, following COMs (Trabucco et al., 2021), we consider the constrained counterpart of Eq. (5):

$$\min_{\theta} \mathbb{E}_{(x,y)\sim P_0}\left[(f_\theta(x)-y)^2\right] \quad \text{s.t.} \quad \Psi_c(\theta) \le \tau, \tag{15}$$

where $\tau$ specifies the maximum tolerated worst-case predictive bias. The Lagrangian of Eq. (15) recovers Eq. (5) up to a constant $-\lambda\tau$ that does not affect the optimization over $\theta$, with $\lambda \ge 0$ now interpreted as the Lagrange multiplier associated with the constraint. Motivated by dual ascent, our implementation adapts $\lambda$ using the sampled finite-step prediction gain:

$$\lambda \leftarrow \left[\lambda + \eta_\lambda\left(\mathcal{L}_{\text{bias}} - \tau\right)\right]_+, \quad \text{where} \quad \mathcal{L}_{\text{bias}} = \frac{1}{B}\sum_{i=1}^{B} f_\theta(\tilde{x}_i) - \frac{1}{B}\sum_{i=1}^{B} f_\theta(x_i), \tag{16}$$

and $\{(x_i, \tilde{x}_i)\}_{i=1}^{B}$ denotes the mini-batch of training samples and their corresponding adversarial perturbations obtained from the inner maximization in Algorithm 1. This rule increases $\lambda$ whenever $\mathcal{L}_{\text{bias}}$ exceeds $\tau$ and decreases it otherwise, subject to projection onto $[0, \infty)$. It targets the sampled finite-step gain and does not by itself certify the constraint $\Psi_c(\theta) \le \tau$.

To verify robustness, we evaluate $\tau \in \{0.2, 0.5, 1, 2, 5\}$ on continuous tasks and $\tau \in \{0.5, 1, 2, 4, 8\}$ on discrete tasks. As shown in Figure 4, normalized scores remain stable across the entire range on both task types, with variations well within the standard error. This confirms that BiaWDRO is robust to the choice of $\tau$ and does not require extensive tuning.

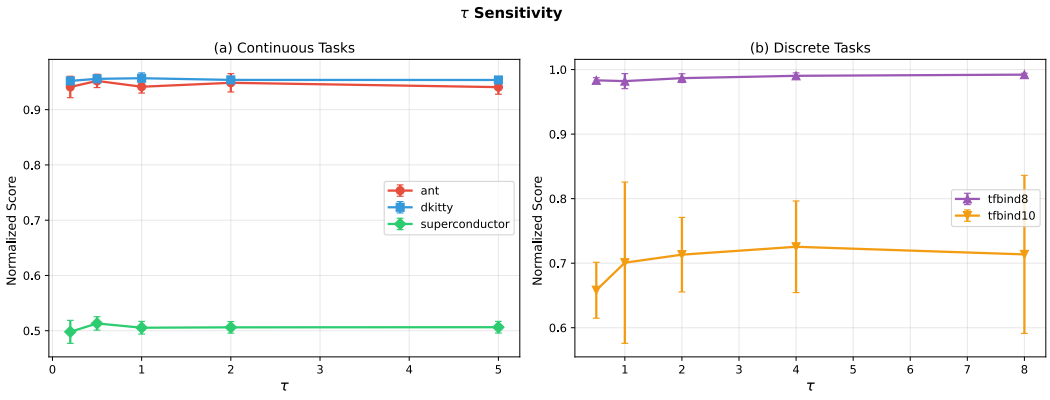



Figure 4: **Sensitivity analysis of $\tau$ on Design-Bench tasks.** (a) Continuous tasks across $\tau \in \{0.2, 0.5, 1, 2, 5\}$. (b) Discrete tasks across $\tau \in \{0.5, 1, 2, 4, 8\}$. Error bars show SE across 8 independent runs. Performance remains stable across a wide range of $\tau$ values, indicating BiaWDRO is robust to this choice.

### C.3 The Influence of Evaluation Budget

Following standard practice in offline MBO (Trabucco et al., 2021), we generate $N = 128$ candidate inputs and evaluate the top-$k$ candidates with the ground truth objective, using the maximum score attained as the performance metric. The value $k$ represents the evaluation budget—the number of candidates we are allowed to test with the true objective function. Performance at small budgets measures the ability to identify high-scoring designs with few oracle evaluations.

Figure 5 compares BiaWDRO with COMs as $k$ varies from 1 to 128. The mean-score curves generally favor BiaWDRO on TFBind8, TFBind10, and Superconductor. On the locomotion tasks (Ant and D'Kitty), both methods improve rapidly at small budgets and then approach a plateau, with BiaWDRO achieving higher mean scores at $k = 128$.

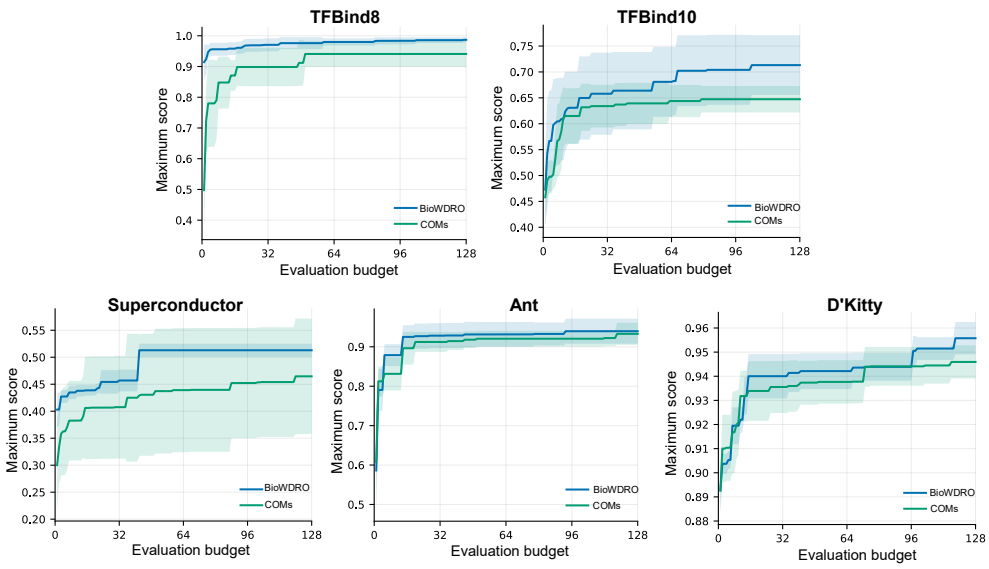



Figure 5: **Evaluation budget analysis across five benchmark tasks.** The x-axis shows how many top candidates ($k$) are evaluated with the ground truth objective. The y-axis shows the maximum score among the evaluated candidates. Curves show the mean and shaded bands show SE across 8 independent runs.

Table 5: **Sensitivity analysis of the Wasserstein radius $\rho$ on Design-Bench tasks.** We report normalized scores (mean $\pm$ SE) over 8 independent runs. Mean rank is computed by ranking each $\rho$ within each task and averaging across tasks. **Blue** denotes the best and **Violet** the second best.

| $\rho$ | Ant | D'Kitty | Superconductor | TFBind8 | TFBind10 | Mean Rank |
|---|---|---|---|---|---|---|
| 0.5 | $0.947 \pm 0.015$ | $0.955 \pm 0.009$ | $0.511 \pm 0.012$ | **$0.989 \pm 0.003$** | $0.679 \pm 0.051$ | **3.2 / 7** |
| 1.0 | **$0.954 \pm 0.016$** | **$0.957 \pm 0.009$** | **$0.513 \pm 0.014$** | $0.984 \pm 0.009$ | $0.658 \pm 0.035$ | **3.2 / 7** |
| 1.5 | **$0.952 \pm 0.012$** | $0.956 \pm 0.007$ | **$0.513 \pm 0.013$** | $0.987 \pm 0.007$ | **$0.713 \pm 0.061$** | **2.2 / 7** |
| 2.0 | $0.939 \pm 0.023$ | **$0.960 \pm 0.007$** | **$0.514 \pm 0.014$** | $0.973 \pm 0.016$ | **$0.708 \pm 0.125$** | 3.4 / 7 |
| 3.0 | $0.941 \pm 0.012$ | $0.955 \pm 0.011$ | $0.511 \pm 0.018$ | **$0.988 \pm 0.006$** | $0.660 \pm 0.039$ | 4.2 / 7 |
| 4.0 | $0.941 \pm 0.012$ | $0.953 \pm 0.008$ | **$0.513 \pm 0.013$** | $0.979 \pm 0.019$ | $0.664 \pm 0.044$ | 4.6 / 7 |
| 5.0 | $0.941 \pm 0.033$ | $0.954 \pm 0.007$ | $0.510 \pm 0.013$ | $0.972 \pm 0.009$ | $0.666 \pm 0.042$ | 5.6 / 7 |

### C.4 Exploration Distance Analysis: Additional Visualizations

This section provides detailed visualizations supplementing the exploration distance analysis in Section 5.2. We present individual trajectory patterns and cumulative distributions to further validate that the exploration difference between COMs and BiaWDRO is systematic across candidates.

**Experimental setup.** For each of the top-128 candidates across all tasks, we have three designs: (1) the initial design sampled from the top-128 designs in the offline dataset, (2) the COMs-optimized design after gradient-based optimization, and (3) the BiaWDRO-optimized design after gradient-based optimization. We compute the distance from the initial design to the COMs-optimized design ($d_{\text{COMs}}$) and from the initial design to the BiaWDRO-optimized design ($d_{\text{BiaWDRO}}$). Distance is measured using Hamming distance for discrete tasks (TFBind8/10, counting the number of differing positions) and Euclidean distance for continuous tasks (Superconductor, Ant, D'Kitty, computing the $\ell_2$ norm in design space). We then calculate the distance difference for each candidate as $\Delta = d_{\text{COMs}} - d_{\text{BiaWDRO}}$, where positive values indicate that COMs moved farther from the initial design than BiaWDRO.

**Individual trajectory patterns.** Figure 6 visualizes within-candidate differences using paired comparisons. For each candidate, we plot its displacement from the initial design under both methods and connect them with a line. This reveals whether the exploration differences reported in Figure 3 are consistent at the individual level. The predominance of upward-sloping orange lines demonstrates that

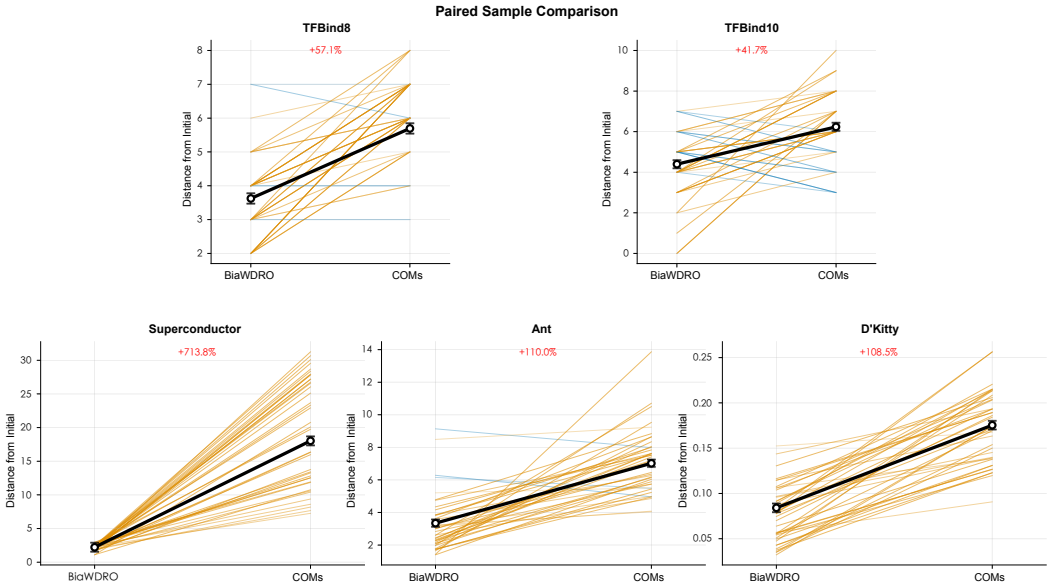



Figure 6: **Paired comparison of exploration distances.** Each line connects the same candidate's distance from initial design under BiaWDRO (left) and COMs (right). Orange: COMs moved farther; blue: BiaWDRO moved farther. Thick black line shows means; percentages indicate relative increases.

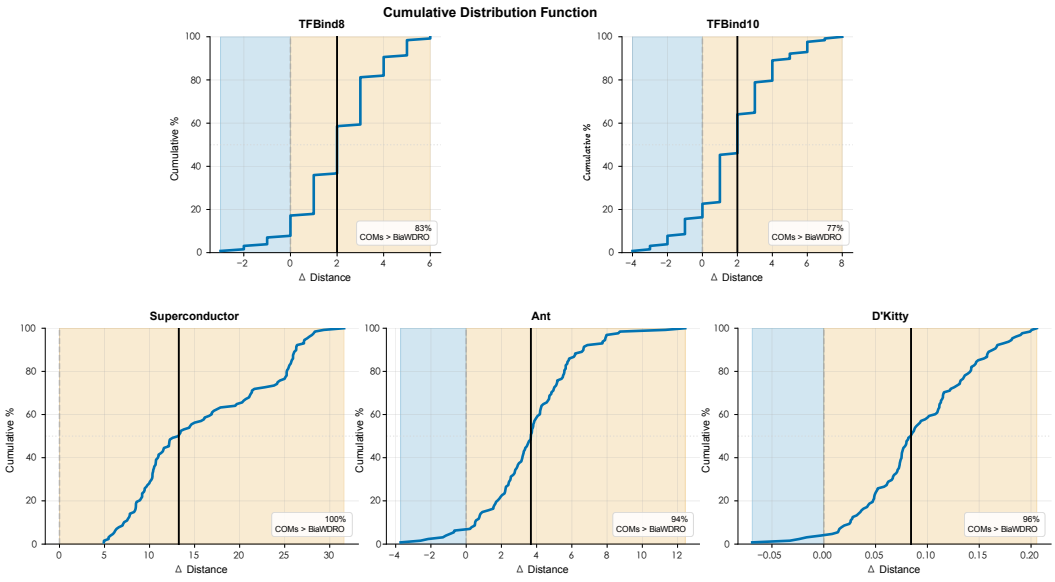



Figure 7: **Cumulative distribution of distance differences.** CDFs show the percentage of candidates with distance differences ($\Delta = d_{\text{COMs}} - d_{\text{BiaWDRO}}$) at or below each x-axis value. Gray dashed line marks equal distances; percentages indicate the proportion where COMs moved farther.

most candidates—not just the average—move farther under COMs, with mean distances increasing by 41.7%–713.8% compared to BiaWDRO. Superconductor exhibits the most dramatic difference (713.8%), while TFBind10 shows the most modest (41.7%). These paired comparisons show that the observed mean-distance difference is shared by many candidates in the evaluated set.

**Cumulative distribution of distance differences.** Figure 7 presents the cumulative distribution of distance differences $\Delta$ across all candidates. Across all tasks, 77–100% of candidates have $\Delta > 0$:

Superconductor achieves complete separation (100%), while discrete tasks range from 77% to 83%. This confirms that the exploration difference is systematic rather than driven by outliers.

Despite smaller displacements from the initial designs, BiaWDRO achieves higher mean scores than COMs (Table 1). This pattern suggests that transport-regularized conservative training can improve design quality while moderating exploration.

### C.5 Sensitivity of Aggregate Rankings

We examine the sensitivity of the aggregate ranking to task inclusion and tie handling using the displayed mean scores in Table 1. All comparisons retain the same 22 methods.

**Task inclusion.** We omit each task in turn and recompute the mean ranks over the remaining four tasks, retaining the original within-task ranks and the convention that ties receive the smallest shared rank. Table 6 reports all five comparisons. BiaWDRO ranks first in four comparisons and second when Superconductor is omitted, where ROOT obtains a mean rank of 3.00 compared with 3.50 for BiaWDRO.

Table 6: **Sensitivity of aggregate rankings to task inclusion.** Each row omits one task from Table 1 and averages the within-task ranks over the remaining four tasks. All 22 methods are retained. Lower mean rank is better; position denotes BiaWDRO's rank among the 22 methods according to this four-task average.

| Omitted task | BiaWDRO mean rank | BiaWDRO position |
|---|---|---|
| TFBind8 | 3.75 | 1 |
| TFBind10 | 3.50 | 1 |
| Superconductor | 3.50 | 2 |
| Ant | 3.00 | 1 |
| D'Kitty | 2.25 | 1 |

**Tie handling.** We also replace the minimum-rank convention for ties with average ranks: methods tied across positions $r$ through $s$ each receive $(r + s)/2$. Under this convention, averaging over all five tasks gives BiaWDRO the best mean rank of 3.3, followed by RaM-ListNet at 5.5 and ROOT at 6.1.

Overall, BiaWDRO remains among the top two methods in all five task-omission comparisons and retains the best mean rank under average-rank tie handling.

## D Proof of the Fitting-Cost and Collapse Results

We first bound the range of training predictions by the regularizer, then compare the fitting risk with the best constant predictor. The proof is given for finite nonnegative convex losses; squared loss yields Theorem 3.1.

### D.1 Transport setting

The dataset is $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, with $n \geq 2$, and the empirical input distribution is $P_0 = n^{-1} \sum_i \delta_{x_i}$, where $\delta_{x_i}$ is the Dirac probability measure at $x_i$. For a fitting loss $\ell$, write $R_n(\theta) = n^{-1} \sum_i \ell(f_\theta(x_i), y_i)$. We use squared loss in the main text and finite nonnegative convex losses for the extensions below.

The input space $\mathcal{X}$ is Polish, and $c : \mathcal{X} \times \mathcal{X} \to [0, \infty)$ is finite, nonnegative and lower semicontinuous, with $c(x, x) = 0$. Fix $\rho > 0$. The surrogate $f_\theta$ is a real-valued Borel function, integrable under the feasible distributions under discussion. Regularizer values used as finite bounds are assumed finite.

Order a transport pair as $(x', x)$, with $x$ its training source. Let $\Pi(P, P_0)$ denote the couplings with first marginal $P$ and second marginal $P_0$, and let $\pi_2$ denote the second marginal of a plan $\pi$. Define

$$W_c(P, P_0) = \inf_{\pi \in \Pi(P,P_0)} \int c(x', x)\, d\pi(x', x), \quad \mathcal{U}_\rho(P_0) = \{P : W_c(P, P_0) \le \rho\}, \tag{D.1}$$

$$\Pi_\rho = \left\{ \pi : \pi_2 = P_0,\ \int c\, d\pi \le \rho \right\}, \qquad \Psi_c(\theta) = \sup_{P \in \mathcal{U}_\rho(P_0)} \{\mathbb{E}_P f_\theta - \mathbb{E}_{P_0} f_\theta\}. \tag{D.2}$$

For fixed marginals, the set of couplings is tight and weakly closed, hence weakly compact. The nonnegative lower-semicontinuous cost integral attains its infimum whenever that infimum is finite. Thus every $P \in \mathcal{U}_\rho(P_0)$ admits a coupling of cost at most $\rho$, and

$$\Psi_c(\theta) = \sup_{\pi \in \Pi_\rho} \int [f_\theta(x') - f_\theta(x)]\, d\pi \ge 0. \tag{D.3}$$

The inequality follows from the feasible stay-put plan. This representation allows us to lower-bound the regularizer by constructing transports between training inputs.

### D.2 Prediction range and the best-constant fitting benchmark

A single training point carries mass $1/n$. To move mass from $x_i$ to $x_j$ without exceeding either that mass or the transport budget, define $v_i = f_\theta(x_i)$ and

$$a_{ij} = \min\left\{ \frac{1}{n}, \frac{\rho}{c(x_j, x_i)} \right\}, \qquad a_* = \min_{i \ne j} a_{ij} > 0, \qquad \rho/0 := +\infty. \tag{D.4}$$

Move mass $a_{ij}$ from the diagonal pair $(x_i, x_i)$ to $(x_j, x_i)$, leaving all other mass in the stay-put plan fixed. The new plan retains source marginal $P_0$, has cost $a_{ij} c(x_j, x_i) \le \rho$, and gains $a_{ij}(v_j - v_i)$. Comparing also with zero gain gives the first inequality below. Choosing indices at the minimum and maximum predictions gives the second:

$$\Psi_c(\theta) \ge a_{ij}[v_j - v_i]_+, \qquad \Delta_v := \max_i v_i - \min_i v_i \le \frac{\Psi_c(\theta)}{a_*}. \tag{D.5}$$

If $\Delta_v = 0$, its bound is immediate. The transport construction remains valid when some training inputs coincide.

Suppose each $\ell(\cdot, y_i)$ is finite, nonnegative and convex on $\mathbb{R}$. Define

$$r_\ell(b) = \frac{1}{n} \sum_i \ell(b, y_i), \qquad R_{\text{const}} = \min_b r_\ell(b), \qquad \mathcal{B}_\ell = \arg\min_b r_\ell(b) \ne \varnothing. \tag{D.6}$$

Fix $b_\ell \in \mathcal{B}_\ell$. Finite convex losses are continuous on $\mathbb{R}$, so convex optimality and the subdifferential sum rule (Boyd et al., 2022) give

$$0 \in \partial r_\ell(b_\ell) = \frac{1}{n} \sum_i \partial_1 \ell(b_\ell, y_i).$$

Hence finite subgradients can be chosen jointly to satisfy

$$s_i \in \partial_1 \ell(b_\ell, y_i), \qquad \sum_i s_i = 0, \qquad \Lambda_\ell = \frac{1}{2na_*} \sum_i |s_i| \ge 0. \tag{D.7}$$

Here $\partial_1$ is the convex subdifferential in the prediction. The balance condition is part of the choice: arbitrary subgradients at nondifferentiable points need not sum to zero. Every balanced choice defines a valid finite $\Lambda_\ell$.

**Proposition D.1** (General convex-loss trade-off)**.** *Suppose each $\ell(\cdot, y_i)$ is finite, nonnegative and convex, and $\mathcal{B}_\ell \ne \varnothing$. With $\Lambda_\ell$ defined in Eq.* (D.7)*,*

$$R_n(\theta) \ge [R_{\text{const}} - \Lambda_\ell \Psi_c(\theta)]_+.$$

*Consequently, $\Psi_c(\theta) \le \tau$ implies $R_n(\theta) \ge [R_{\text{const}} - \Lambda_\ell \tau]_+$ for $\tau \ge 0$.*

*Proof.* The supporting-line inequality yields

$$\ell(v_i, y_i) \geq \ell(b_\ell, y_i) + s_i(v_i - b_\ell).$$

Set $m_0 = (\max_i v_i + \min_i v_i)/2$. Balance eliminates the common offset, while $|v_i - m_0| \leq \Delta_v/2$. Consequently,

$$\begin{aligned} R_n(\theta) &\geq R_{\text{const}} + \frac{1}{n}\sum_i s_i(v_i - b_\ell) = R_{\text{const}} + \frac{1}{n}\sum_i s_i(v_i - m_0) \\ &\geq R_{\text{const}} - \frac{\Delta_v}{2n}\sum_i |s_i| \geq R_{\text{const}} - \Lambda_\ell \Psi_c(\theta). \end{aligned} \tag{D.8}$$

Combining this with nonnegativity of the loss proves, for every $\tau \geq 0$,

$$\Psi_c(\theta) \leq \tau \quad \Longrightarrow \quad R_n(\theta) \geq [R_{\text{const}} - \Lambda_\ell \tau]_+. \tag{D.9}$$

This proves the proposition. □

The bound limits how much fitting can improve over the best constant while retaining a prescribed predictive-bias tolerance.

### D.3 A FINITE PENALTY FORCES EXACT TRAINING-SUPPORT COLLAPSE

Assume the model class contains a parameter $\theta_{\text{const}}$ with $f_{\theta_{\text{const}}}(x) = b_0$ for every $x \in \mathcal{X}$, for some $b_0 \in \mathcal{B}_\ell$. Its regularizer is zero and $J_\lambda(\theta_{\text{const}}) = R_{\text{const}}$.

**Proposition D.2** (General convex-loss collapse)**.** *Under the assumptions of Proposition D.1, suppose the model class contains the globally constant comparator described above. If $\lambda > \Lambda_\ell$, every global minimizer of $J_\lambda$ has $\Psi_c(\theta) = 0$, $R_n(\theta) = R_{\text{const}}$, and $f_\theta(x_i) = b$ for every $i$, for some $b \in \mathcal{B}_\ell$.*

*Proof.* For any $\theta$ with finite objective, (D.8) implies

$$J_\lambda(\theta) = R_n(\theta) + \lambda \Psi_c(\theta) \geq R_{\text{const}} + (\lambda - \Lambda_\ell)\Psi_c(\theta). \tag{D.10}$$

For $\lambda > \Lambda_\ell$, the right side is at least $R_{\text{const}}$. The globally constant comparison function attains this value, so the minimum exists and equals $R_{\text{const}}$.

At any global minimizer $\theta_\lambda$, equality of the objective value forces $\Psi_c(\theta_\lambda) = 0$ because $\lambda - \Lambda_\ell > 0$. The range bound (D.5) then implies $f_{\theta_\lambda}(x_i) = b$ for every $i$, for some common $b$. Since its fitting risk is $J_\lambda(\theta_\lambda) = R_{\text{const}}$, this common value satisfies $r_\ell(b) = R_{\text{const}}$, hence $b \in \mathcal{B}_\ell$. This proves all assertions. □

For a general convex loss the best constant need not be unique. The conclusion concerns the training predictions of global minimizers; it does not specify all off-support values.

### D.4 SQUARED-LOSS SPECIALIZATION AND FURTHER CONSEQUENCES

*Proof of Theorem 3.1.* For $\ell(u, y_i) = (u - y_i)^2$, the unique best constant is $b_\ell = \bar{y} = n^{-1}\sum_i y_i$, with $R_{\text{const}} = n^{-1}\sum_i (y_i - \bar{y})^2$. Choose $s_i = 2(\bar{y} - y_i)$; then $\sum_i s_i = 0$ and

$$\Lambda_\ell = \frac{1}{2na_*}\sum_i |2(\bar{y} - y_i)| = \frac{1}{na_*}\sum_i |y_i - \bar{y}| = \Lambda.$$

Proposition D.1 gives Eq. (14), and Proposition D.2 gives the stated collapse result. □

**Approximate solutions.** If $\lambda > \Lambda_\ell$ and $J_\lambda(\widetilde{\theta}) \leq R_{\text{const}} + \xi$, $\xi \geq 0$, retaining the last term of (D.10) yields

$$\Psi_c(\widetilde{\theta}) \leq \frac{\xi}{\lambda - \Lambda_\ell}, \qquad \text{range}_i\, f_{\widetilde{\theta}}(x_i) \leq \frac{\xi}{a_*(\lambda - \Lambda_\ell)}. \tag{D.11}$$

Thus near-optimal penalized objectives quantitatively restrict the remaining training-prediction variation.

**The hard-constraint limit.** For $V_\ell(\tau) = \inf\{R_n(\theta) : \Psi_c(\theta) \le \tau\}$, the same constant comparison function is feasible for every $\tau \ge 0$. Therefore

$$[R_{\text{const}} - \Lambda_\ell \tau]_+ \le V_\ell(\tau) \le R_{\text{const}}, \qquad \lim_{\tau \downarrow 0} V_\ell(\tau) = R_{\text{const}}. \tag{D.12}$$

The lower bound is (D.9), and the limit follows by squeezing. For squared loss, this limit is the empirical mean squared deviation of the labels from their mean.

**Why the general theorem uses a strict penalty threshold.** Convexity gives only the linear remainder in (D.10); at $\lambda = \Lambda_\ell$, it need not force zero bias. If each $\ell(\cdot, y_i)$ is additionally $\mu_\ell$-strongly convex, $\mu_\ell > 0$, the stronger supporting inequality instead gives

$$J_\lambda(\theta) \ge R_{\text{const}} + \frac{\mu_\ell}{2n} \sum_i (f_\theta(x_i) - b_\ell)^2 + (\lambda - \Lambda_\ell)\Psi_c(\theta). \tag{D.13}$$

This follows by adding the strong-convexity remainder to each supporting line and repeating (D.8). With $\lambda \ge \Lambda_\ell$ and $\lambda > 0$, equality at a global minimum then forces all training predictions to equal the unique best constant $b_\ell$; substituting into $J_\lambda = R_{\text{const}}$ also gives $\Psi_c(\theta) = 0$. The main theorem uses the strict sufficient threshold to match the general convex-loss statement.

# E Detailed Related Work

## E.1 Offline Model-Based Optimization

Offline MBO aims to find a design $x$ that maximizes an unknown objective function $f(x)$ using only a static offline dataset. The core challenge lies in preventing OOD overestimation during optimization. Existing methods address this challenge from several perspectives.

**Conservative Methods.** These methods explicitly constrain model behavior on OOD regions to prevent overestimation of predicted performance. COMs (Trabucco et al., 2021) penalizes surrogate model predictions on OOD inputs that are discovered through gradient ascent optimization. NEMO (Fu & Levine, 2021) bounds the prediction uncertainty of the surrogate model through normalized maximum likelihood estimation. BDI (Chen et al., 2022) introduces a backward mapping regularization technique to ensure that generated designs remain close to the original training data distribution. PGS (Chemingui et al., 2024) reformulates offline MBO as an offline RL problem and applies conservative Q-learning techniques to address distributional shift. GABO (Yao et al., 2024) introduces adaptive source critic regularization (aSCR) to dynamically constrain optimization trajectories within the in-distribution region via a Wasserstein-based critic. DynAMO (Yao et al., 2025) further penalizes the MBO objective with a KL divergence term to match a temperature-weighted offline distribution, while using an adversarial source critic to constrain OOD evaluation.

**Generative and Inverse Approaches.** Rather than optimizing against a proxy, these methods learn the conditional distribution $p(\mathbf{x}|y)$ to directly generate high-scoring designs. CbAS (Brookes et al., 2019) trains VAEs and iteratively re-weights the density towards high-scoring regions. MINs (Kumar & Levine, 2020) uses Conditional GANs to model the inverse mapping from scores to designs. Recent works leverage diffusion models: DDOM (Krishnamoorthy et al., 2023) employs conditional diffusion for high-fidelity generation, while RGD (Chen et al., 2024) introduces proxy-enhanced sampling with a refinement module. ROOT (Dao et al., 2025) reframes offline MBO as a distributional translation task, learning a probabilistic bridge that maps low-value designs to high-value ones using synthetic pairs generated from Gaussian process posteriors.

**Gradient and Geometric Regularization.** These methods regularize the surrogate's loss landscape rather than directly penalizing OOD values. RoMA (Yu et al., 2021) enforces local smoothness through a two-stage adaptation process. IGNITE (Dao et al., 2024) constrains gradient norms to reduce loss sharpness. Match-OPT (Hoang et al., 2025) uses gradient matching via line integrals to align surrogate gradients with the true objective function.

**Ranking and Collaborative Learning.** RaM (Tan et al., 2025) replaces MSE with learning-to-rank objectives for improved optimization guidance. ICT (Yuan et al., 2023) and Tri-Mentoring (Chen et al., 2023) leverage co-teaching frameworks where multiple surrogate proxies exchange pseudo-labels or ranking signals to better handle OOD data.

**Trajectory-Based Optimization.** BONET (Mashkaria et al., 2023) generates optimization trajectories using autoregressive models to capture sequential design improvements. GTG (Yun et al., 2024) improves this with locality-biased synthetic trajectories and conditional diffusion.

### E.2 Distributionally Robust Optimization

DRO optimizes against the worst-case distribution within an ambiguity set, providing a principled framework for handling distributional uncertainty. Pioneered by Scarf (1958), early work explored moment-based (Delage & Ye, 2010; Wiesemann et al., 2014) and $\phi$-divergence-based ambiguity sets, including KL divergence (Ben-Tal et al., 2013) and $\chi^2$ divergence (Namkoong & Duchi, 2016). However, $\phi$-divergence-based methods require absolute continuity with respect to the nominal distribution, limiting their applicability. The Wasserstein distance, rooted in optimal transport theory, circumvents this limitation by providing a valid metric between discrete empirical and continuous population distributions. Mohajerin Esfahani & Kuhn (2018) showed that worst-case expectations over Wasserstein balls admit finite convex reformulations, while Blanchet & Murthy (2019) established statistical guarantees linking the radius to convergence rates. Building upon that theoretical framework, Sinha et al. (2017) extended WDRO to non-convex objectives via Lagrangian relaxation, enabling its application to deep learning. Other work has explored connections between WDRO and regularization (Gao et al., 2024) as well as robust classification (Ho-Nguyen & Wright, 2023).