
DeepDRK: Deep Dependency Regularized Knockoff for Feature Selection

Hongyu Shen¹ Yici Yan² Zhizhen Zhao¹
Department of Electrical and Computer Engineering¹,
University of Illinois at Urbana Champaign.
Department of Statistics²,
University of Illinois at Urbana Champaign.
{hongyu2, yiciyan2, zhizhenz}@illinois.edu

Abstract

Model-X knockoff has garnered significant attention among various feature selection methods due to its guarantees for controlling the false discovery rate (FDR). Since its introduction in parametric design, knockoff techniques have evolved to handle arbitrary data distributions using deep learning-based generative models. However, we have observed limitations in the current implementations of the deep Model-X knockoff framework. Notably, the “swap property” that knockoffs require often faces challenges at the sample level, resulting in diminished selection power. To address these issues, we develop “Deep Dependency Regularized Knockoff (DeepDRK),” a distribution-free deep learning method that effectively balances FDR and power. In DeepDRK, we introduce a novel formulation of the knockoff model as a learning problem under multi-source adversarial attacks. By employing an innovative perturbation technique, we achieve lower FDR and higher power. Our model outperforms existing benchmarks across synthetic, semi-synthetic, and real-world datasets, particularly when sample sizes are small and data distributions are non-Gaussian.

1 Introduction

Feature selection (FS) has garnered significant attention over the past few decades due to the rapidly increasing dimensionality of data, as well as the associated computational, storage, and noise challenges [22, 15]. Successfully identifying the true informative features among inputs can significantly enhance the performance of analysis frameworks and drive advancements in fields such as biology, neuroscience, medicine, economics, and social sciences [15, 55]. However, the task of accurately pinpointing informative features is often considered nearly impossible, particularly due to the limited availability of data relative to the increasing dimensionality [55, 11]. A practical approach to address this challenge is the development of algorithms designed to select features while maintaining controlled error rates. Targeting this goal, Model-X knockoffs, a novel framework, is proposed in [4, 11] to select relevant features while controlling the false discovery rate (FDR). In contrast to the classical setup, where assumptions on the correlations between input features and the response are imposed [6, 19], the Model-X knockoff framework only requires a linear relationship between the response and the features. With a strong finite-sample FDR guarantee, Model-X knockoff saw broad applications in domains such as biology, neuroscience, and medicine, where the sample size is limited [4, 11, 27, 47, 55, 38, 62].

There have been considerable developments of knockoffs since its debut. In scenarios where feature distributions are complex, various deep learning methods [27, 47, 55, 38, 62] have been proposed. However, we observe major limitations despite improved performance. First, the performance of existing methods varies across different data distributions. Second, the quality of selection declines

when the sample size is relatively small. Third, training the deep knockoff generation models can be challenging due to competing loss terms in the training objective. We elaborate on the drawbacks in Sections 2.2 and 3.2.

In this paper, we address these issues by proposing the Deep Dependency Regularized Knockoff (DeepDRK), a deep learning-based pipeline. We formulate the knockoff generation as an adversarial attack problem involving multiple sources. By optimizing a model against the adversarial environments, we achieve better “swap property” [4] compared to the baseline models. DeepDRK is also equipped with a novel perturbation technique to reduce “reconstructability” [54], which in turn controls FDR and boosts selection power. The experiments conducted on synthetic, semi-synthetic, and real datasets demonstrate that our pipeline outperforms existing methods across various scenarios.

2 Background and Related Works

2.1 Model-X Knockoffs for FDR control

The Model-X knockoffs framework consists of two main components. Given the explanatory variables $X = (X_1, X_2, \dots, X_p)^\top \in \mathbb{R}^p$ and the response variable Y (Y continuous for regression and categorical for classification), the framework requires: 1. a knockoff $\tilde{X} = (\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_p)^\top$ that “fakes” X ; 2. the knockoff statistics w_j for $j \in [p]$ that assess the importance of each feature X_j . The knockoff $\tilde{X}$ is required to be independent of Y conditioning on X , and must satisfy the swap property:

$$(X, \tilde{X})_{\text{swap}(B)} \stackrel{d}{=} (X, \tilde{X}), \quad \forall B \subset [p]. \quad (1)$$

Here $\text{swap}(B)$ exchanges the positions of any variable $X_j, j \in B$, with its knockoff $\tilde{X}_j$. The knockoff statistic $w_j((X, \tilde{X}), Y)$ (for $j \in [p]$) depends on the concatenated variable $(X, \tilde{X})$ and Y and must satisfy the flip-sign property:

$$w_j((X, \tilde{X})_{\text{swap}(B)}, Y) = \begin{cases} w_j((X, \tilde{X}), Y) & \text{if } j \notin B \\ -w_j((X, \tilde{X}), Y) & \text{if } j \in B \end{cases} \quad (2)$$

The functions $w_j(\cdot)$ with $j \in [p]$ have many candidates, for example $w_j = |\beta_j| - |\tilde{\beta}_j|$, where β_j and $\tilde{\beta}_j$ are the corresponding regression coefficients of X_j and $\tilde{X}_j$ with the regression function $Y = \sum_{j=1}^p (X_j \beta_j + \tilde{X}_j \tilde{\beta}_j) + \epsilon$, where ϵ is independently drawn from the normal distribution.

When the two knockoff conditions (i.e., Eq. (1) and (2)) are met, one can select features by $\mathcal{S} = \{w_j \geq \tau_q\}$, where

$$\tau_q = \min_{t \geq 0} \left\{ t : \frac{1 + |\{j : w_j \leq -t\}|}{\max(1, |\{j : w_j \geq t\}|)} \leq q \right\}. \quad (3)$$

To assess the feature selection quality, FDR is commonly used as an average Type I error of selected features [4], which is defined as follows. Let $\mathcal{S} \subset [p]$ be an arbitrary set of selected indices, and let β^* denote the underlying true regression coefficients. The FDR for the selection $\mathcal{S}$ is

$$\text{FDR} = \mathbb{E} \left[\frac{\#\{j : \beta_j^* = 0 \text{ and } j \in \mathcal{S}\}}{\#\{j : j \in \mathcal{S}\} \vee 1} \right] \quad (4)$$

The control of FDR is guaranteed by the following theorem from [11]:

Theorem 2.1. *Given the knockoff that satisfies the swap property in Eq. (1), the knockoff statistic that satisfies Eq. (2), and $\mathcal{S} = \{w_j \geq \tau_q\}$, we have $\text{FDR} \leq q$.*

2.2 Related Works

Model-X knockoff is first studied under Gaussian design. Namely, the original variable $X \sim \mathcal{N}(\mu, \Sigma)$ with μ and Σ known. Since Gaussian design does not naturally generalize to complex data distributions, several methods are proposed to weaken the assumption. Among them, model-specific ones

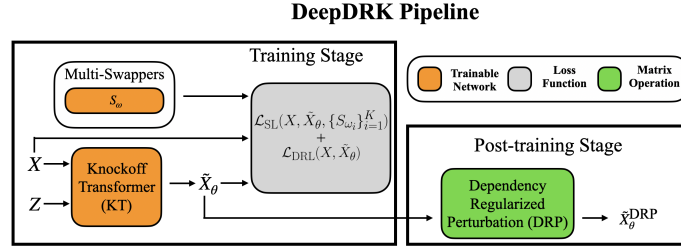


Figure 1: The illustration of the DeepDRK pipeline, which consists of two components: 1. the training stage that optimizes the knockoff Transformer and swappers by $\mathcal{L}_{SL}$ and $\mathcal{L}_{DRL}$; 2. the post-training stage that generates the knockoff $\tilde{X}^{DRP_\theta}$ via dependency regularized perturbation.

such as AEknockoff [34], Hidden Markov Model (HMM) knockoff [51], and MASS [20] all propose parametric alternatives to Gaussian design. These methods can better learn the data distribution, while keeping the sampling process relatively simple. Nevertheless, they pose assumptions to the design distribution, which can be problematic if actual data does not coincide. To gain further flexibility, various deep-learning-based models are developed to generate knockoffs from distributions beyond parametric setup. DDLK [55] and sRMMD [38] utilize different metrics to measure the distances between the original and the knockoff covariates. They apply different regularization terms to impose the “swap property”. He et al. [24] introduced a KnockoffScreen procedure to generate multiple knockoffs to improve the stability by minimizing the variance during knockoff construction. KnockoffGAN [27] and Deep Knockoff [47] take advantage of the deep learning structures to create likelihood-free generative models for the knockoff generation.

Despite the flexibility to learn the data distribution, deep-learning-based models suffer from a major drawback. Knockoff generations based on distribution-free sampling methods such as generative adversarial networks (GAN) [21, 3] tend to overfit, namely to learn the data X exactly. The reason is that the notion of swap property for continuous distributions is not well defined at the sample level. To satisfy the swap property, one needs to independently sample $\tilde{X}_j$ from the conditional law $P_{X_j}(\cdot | X_{-j})$, where X_{-j} denotes the vector $(X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_p)$. At the sample level, each realization of $X_{-j} = x_{-j}^i$ is almost surely different and only associates to one corresponding sample $X_j = x_j^i$, causing the conditional law to degenerate to sum of Diracs. As a result, minimizing the distance between $(X, \tilde{X})$ and $(X, \tilde{X})_{\text{swap}(B)}$ will push $\tilde{X}$ towards X and introduce high collinearity that makes the feature selection powerless, i.e., with high type II error. To tackle this issue, DDLK [55] suggests an entropic regularization. Yet it still lacks power and is computationally expensive.

2.3 Boost Power by reducing reconstructability

The issue of lacking power in the knockoff selection is solved in the Gaussian case [54]. Assuming the knowledge of both mean and covariance of $X \sim \mathcal{N}(\mu, \Sigma)$, the swap property is easily satisfied by setting $\tilde{X}_j \sim \mathcal{N}(\mu_j, \Sigma_{jj})$ and $\Sigma_{ij} = \text{Var}(X_i, \tilde{X}_j)$, for $i \neq j$, $i, j \in [p]$. Barber & Candès [4] originally propose to minimize $\text{Var}(X_j, \tilde{X}_j)$ for all $j \in [p]$ using semi-definite programming (SDP), to prevent $\tilde{X}_j$ to be highly correlated with X_j . However, Spector & Janson [54] observed that the SDP knockoff still lacks feature selection power, as merely decorrelating X_j and $\tilde{X}_j$ is not enough, and $(X, \tilde{X})$ can still be (almost) linearly dependent in various cases. This is referred to as high reconstructability¹ in their paper, which can be considered as a population counterpart of collinearity (see Appendix A for more details). To tackle the problem, [54] proposed to maximize the expected conditional variance $\mathbb{E}\text{Var}(X_j | X_{-j}, \tilde{X})$, which admits close-form solution whenever X is Gaussian.

3 Method

DeepDRK in Figure 1 provides a novel way to generate knockoff $\tilde{X}$ while reducing the reconstructability (see Section 2.3) between the generated knockoff $\tilde{X}$ and the input X for data with

¹The “reconstructability” describes how easy a variable X_j can be constructed deterministically from X_{-j} and $\tilde{X}$ [54].

complex distributions. The generated knockoff can then be used to perform FDR-controlled feature selection following the Model-X knockoff framework (see Section 2.1). Overall, DeepDRK contains two main components. It first trains a transformer-based deep learning model, referred to as Knockoff Transformer (KT), to obtain the swap property and reduce the reconstructability of the generated knockoff. This is achieved by incorporating adversarial attacks with multi-swappers. Secondly, a dependency regularized perturbation technique (DRP) is developed to further reduce the reconstructability for $\tilde{X}$ post training. We will elaborate on these two components in the following two subsections. In this section, we slightly abuse the notation such that X and $\tilde{X}$ also denote the corresponding data matrices.

3.1 Training with Knockoff Transformer and Swappers

The KT aims to generate knockoffs, denoted by $\tilde{X}_\theta$, which are parameterized by a transformer network with parameters θ . The loss for training the KT contains a swap loss (SL) $\mathcal{L}_{\text{SL}}$, which enforces the swap property, and a dependency regularization loss (DRL) $\mathcal{L}_{\text{DRL}}$, which controls the reconstructability of the knockoff. K different neural network parameterized swappers, denoted by $\{S_{\omega_i}\}_{i=1}^K$, are used to test whether the generated knockoffs satisfy the swap property. Thus, the KT is trained adversarially according to the following objective,

$$\min_{\theta} \max_{\omega_1, \dots, \omega_K} \{\mathcal{L}_{\text{SL}}(X, \tilde{X}_\theta, \{S_{\omega_i}\}_{i=1}^K) + \mathcal{L}_{\text{DRL}}(X, \tilde{X}_\theta)\}. \quad (5)$$

Note that we use $\tilde{X}_\theta$ and $\tilde{X}$ interchangeably; however, the former emphasizes that the knockoff depends on the model weights θ . We discuss each loss function below. Details on the network architectures for KT and swappers are deferred to Appendix B.1. The training algorithm is in Appendix B.2.

3.1.1 Swap Loss

The swap loss is designed to enforce the swap property and is defined as follows:

$$\begin{aligned} \mathcal{L}_{\text{SL}}(X, \tilde{X}_\theta, \{S_{\omega_i}\}_{i=1}^K) &= \frac{1}{K} \sum_{i=1}^K \text{SWD}((X, \tilde{X}_\theta), (X, \tilde{X}_\theta)_{S_{\omega_i}}) \\ &+ \lambda_1 \cdot \text{REx}(X, \tilde{X}_\theta, \{S_{\omega_i}\}_{i=1}^K) + \lambda_2 \cdot \mathcal{L}_{\text{swapper}}(\{S_{\omega_i}\}_{i=1}^K), \end{aligned} \quad (6)$$

where λ_1 and λ_2 are hyperparameters.

The first term in Eq. (6) uses the sliced Wasserstein distance (SWD, see Appendix C for definition) to measure the distance between a pair of joint distributions for $(X, \tilde{X}_\theta)$ and $(X, \tilde{X}_\theta)_{S_{\omega_i}}$, where the swapper is parameterized by ω_i . We sum over K SWDs computed for the distributions modified under different swappers to capture the effects of multiple swap attacks. We utilize SWD to compare distributions because it excels in handling complex data distributions and is computationally efficient [13, 28, 14].

Sudarshan et al. [55] introduced a single swap attack parameterized by a neural network. However, we observe that minimizing the worst case swap for all $B \subset [p]$, as suggested by [55], cannot guarantee the swap property of $\tilde{X}_\theta$. To address this limitation, we introduce a “multi-swapper” setup that uses multiple swappers to enforce the swap property. And this leads to the introduction of the second and the third terms in the objective in Eq. (6).

The second term in Eq. (6), $\text{REx}(X, \tilde{X}_\theta, \{S_{\omega_i}\}_{i=1}^K)$ evaluates the variance of the sliced Wasserstein distances $\text{SWD}((X, \tilde{X}_\theta), (X, \tilde{X}_\theta)_{S_{\omega_i}})$ with K realizations of ω [30]. When $\text{REx}(X, \tilde{X}_\theta, \{S_{\omega_i}\}_{i=1}^K) = 0$, the sliced Wasserstein distances are identical across all swappers. Therefore, complementary to the first term, minimizing this term improves the adherence of swap property for the generated $\tilde{X}_\theta$.

The third term in Eq. (6) is introduced to avoid mode collapse on the parameters ω_i of different swappers and ensure each swapper characterizes a different adversarial environment:

$$\mathcal{L}_{\text{swapper}}(\{S_{\omega_i}\}_{i=1}^K) = \frac{1}{|C|} \sum_{(i,j) \in C} \text{sim}(S_{\omega_i}, S_{\omega_j}), \quad (7)$$

where $C = \{(i, j) | i, j \in [K], i \neq j\}$, and $\text{sim}(\cdot, \cdot)$ is the cosine similarity between the weights ω of a pair of different swappers. Without this regularization, all swappers could collapse to a single mode such that the multi-swapper scheme reduces to a single-swapper setup.

Overall, the swap loss $\mathcal{L}_{\text{SL}}$ enforces the swap property via the novel multi-swapper design. Such design provides a more robust assurance of the swap property through multiple adversarial swap attacks, which is shown in the ablation studies in Appendices I.2 and I.3.

3.1.2 Dependency Regularization Loss

As discussed in Section 2.2, pursuing the swap property at the sample level often leads to severe overfitting of $\tilde{X}_\theta$, i.e., pushing $\tilde{X}_\theta$ towards X , which results in high collinearity in feature selection. To address this, the DRL is introduced to reduce the reconstructability between X and $\tilde{X}$:

$$\mathcal{L}_{\text{DRL}}(X, \tilde{X}_\theta) = \lambda_3 \cdot \text{SWC}(X, \tilde{X}_\theta), \quad (8)$$

where λ_3 is a hyperparameter. The SWC term in Eq. (8) refers to the sliced Wasserstein correlation [33], which quantitatively measures the dependency between two random vectors in the same space. More specifically, let Z_1 and Z_2 be two p -dimensional random vectors. $\text{SWC}(Z_1, Z_2) = 0$ indicates that Z_1 and Z_2 are independent, while $\text{SWC}(Z_1, Z_2) = 1$ suggests a linear relationship between each other (see Appendix D for more details on SWC). In DeepDRK, we minimize SWC to reduce the reconstructability, a procedure similar to [54]. The intuition is as follows. If the joint distribution of X is known, then for each $j \in [p]$, the knockoff $\tilde{X}_j$ should be sampled from $P_j(\cdot | X_{-j})$, making X_j and $\tilde{X}_j$ less dependent. In such case the swap property is ensured, and collinearity/reconstructability is reduced due to independence. As we do not have access to the joint law, we want the variables to be less dependent. Since collinearity exists within X , merely decorrelate X_j and $\tilde{X}_j$ is not enough. Thus, we minimize SWC to reduce the dependence between X and $\tilde{X}$. We refer readers to Appendix A for more discussions.

3.2 Dependency Regularization Perturbation

Empirically we observe a competition between $\mathcal{L}_{\text{SL}}$ and $\mathcal{L}_{\text{DRL}}$ in Eq. (5), which adds difficulty to the training procedure. Specifically, the $\mathcal{L}_{\text{SL}}$ is dominating and the $\mathcal{L}_{\text{DRL}}$ increases quickly after a short decreasing period. We are the first to observe this phenomenon in all deep-learning based knockoff generation models when one tries to gain power [47, 55, 38, 27]. We include the experimental evidence in Appendix E. We suggest the following explanation: minimizing the swap loss, which corresponds to FDR control, is the same as controlling Type I error. Similarly, minimizing the dependency loss is to control Type II error. With a fixed number of observations, it is well known that Type I error and Type II error can not decrease at the same time after reaching a certain threshold. In the framework of model-X knockoff, we aim to boost as much power as possible given the FDR is controlled at a certain level, a similar idea as the uniformly most powerful (UMP) test [12]. For this reason, we propose DRP as a post-training technique to further boost power.

DRP is a sample-level perturbation that further eliminates dependency between X and the knockoff. More specifically, DRP perturbs the generated $\tilde{X}_\theta$ with the row-permuted version of X , denoted as X_{rp} . After applying DRP, the final knockoff $\tilde{X}_{\theta,n}^{\text{DRP}}$ becomes:

$$\tilde{X}_{\theta,n}^{\text{DRP}} = (1 - \alpha_n) \cdot \tilde{X}_\theta + \alpha_n \cdot X_{\text{rp}}, \quad (9)$$

where α_n is a preset perturbation weight, n is the sample size, and $\alpha_n \rightarrow 0$ when $n \rightarrow \infty$. In the following, we remove n or θ whenever the context is clear. $\tilde{X}^{\text{DRP}}$ has a smaller SWC with X , since X_{rp} is independent of X . Despite the perturbation increases the swap loss, such impact is negligible when the sample size is large. More specifically, we present the following Lemma 3.1 and Proposition 3.2. Let $\hat{P}_n$ and $\hat{P}_n^B$ denote the empirical joint distribution of the sample X and $\tilde{X}$ before and after the swap: $(X, \tilde{X}) \sim \hat{P}_n$, $(X, \tilde{X})_{\text{swap}(B)} \sim \hat{P}_n^B$. And P and P^B denote their corresponding population distributions.

Lemma 3.1. *Under mild conditions[41], the slice Wasserstein distance between the empirical distributions of $(X, \tilde{X})$ and $(X, \tilde{X})_{\text{swap}(B)}$ and their corresponding population distributions is of scale $\mathcal{O}(n^{-1/2})$, i.e., $\text{SWD}(\hat{P}_n, \hat{P}_n^B) = \text{SWD}(P, P^B) + \mathcal{O}(n^{-1/2})$.*

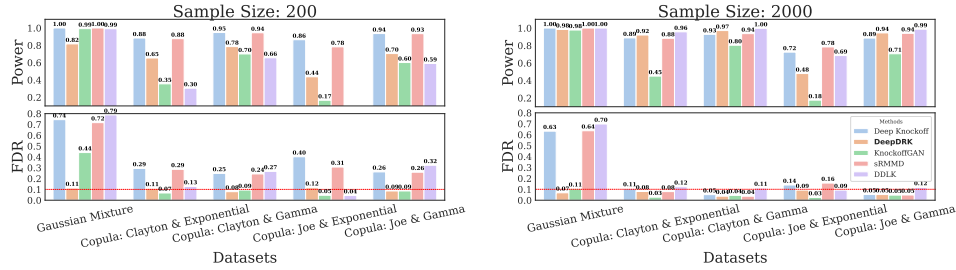


Figure 2: Power and FDR for different knockoff models on the synthetic datasets with $\beta \sim \frac{p}{15 \cdot \sqrt{N}} \cdot \text{Rademacher}(0.5)$. The red horizontal line indicates the 0.1 FDR threshold.

Proposition 3.2. Let $\alpha_n \lesssim \mathcal{O}(n^{-1/2})$ in Eq. (9), and denote $(X, \tilde{X}^{DRP}) \sim \hat{P}_{n,DRP}$, $(X, \tilde{X}^{DRP})_{\text{swap}(B)} \sim \hat{P}_{n,DRP}^B$. Then $\text{SWD}(\hat{P}_{n,DRP}, \hat{P}_{n,DRP}^B) = \text{SWD}(P, P^B)$ when $n \rightarrow \infty$.

The proofs can be found in Appendix F. Lemma 3.1 suggests a general case where the swap loss enforced at the sample level differs from that of the population level from a term of scale $n^{-1/2}$, which vanishes when $n \rightarrow \infty$. Proposition 3.2 provides the rationale on the proposal of the perturbation technique in Eq. (9), such that the DRP term is negligible asymptotically. Besides the theoretical justification, we also empirically show in Appendix G that the perturbation is beneficial.

4 Experiment

We assess the performance of DeepDRK against several benchmark models under three experimental setups: 1. a fully synthetic setup where both the input variables and the response variable follow predefined distributions; 2. a semi-synthetic setup, in which the input variables are derived from real-world datasets and the response variable is generated based on known relationships with the inputs; and 3. feature selection (FS) using a real-world dataset. The experiments are designed to encompass a range of datasets with varying p/n ratios and distributions of X , aiming to provide a comprehensive evaluation of model performance. The benchmark models are Deep Knockoff [47]², DDLK [55]³, KnockoffGAN [27]⁴ and sRMMD [38]⁵, with links of the code implementation listed in the footnote. The implementation details for the training, feature selection, and data preparation are available in Appendix H. In the following, we describe the datasets and the associated experimental results. We also consider the ablation study to illustrate the benefits obtained by having the following proposed terms: REx , $\mathcal{L}_{\text{swapper}}$, the multi-swapper setup, and the dependency regularization perturbation. Empirically, these terms help to improve power and control the FDR. Due to space limitation, we defer details for the ablation studies to Appendix I.2 and I.3. DeepDRK is implemented in PyTorch [45] and is accessible at: <https://github.com/nowonder2000/DeepDRK>.

4.1 The Synthetic Experiments

To properly evaluate the performance, we follow a well-designed experimental setup by [55, 38] to generate different datasets specified by (X, Y) . Here $X \in \mathbb{R}^p$ is the collection dependent variables that follows pre-defined distributions. $Y \in \mathbb{R}$ is the response variable that is modeled as $Y \sim \mathcal{N}(X^T \beta, 1)$. The underlying true β is a p -dimensional vector, where each entry is drawn independently from the distribution $\frac{p}{15 \cdot \sqrt{n}} \cdot \text{Rademacher}(0.5)$.

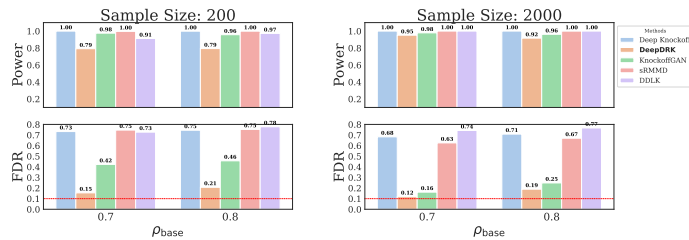


Figure 3: Power and FDR for different knockoff models on the mixture of Gaussian data on different ρ_{base} setups. The red horizontal line indicates the 0.1 FDR threshold. This figure is complementary to Figure 2 for including two additional Gaussian mixture data with higher ρ_{base} values.

²<https://github.com/msesia/deepknockoffs>

³<https://github.com/rajesh-lab/ddlk>

⁴<https://github.com/vanderschaarlab/mlforhealthlabpub/tree/main/alg/knockoffgan>

⁵<https://github.com/ShoaibBinMasud/soft-rank-energy-and-applications>

n	Method	FDR					Power				
		mean	std	median	5% quantile	95% quantile	mean	std	median	5% quantile	95% quantile
200	DDLK	0.772	0.025	0.781	0.726	0.791	0.971	0.033	0.983	0.911	0.995
	Deep Knockoff	0.735	0.028	0.740	0.692	0.769	0.999	0.001	0.999	0.997	1.000
	KnockoffGAN	0.390	0.110	0.396	0.230	0.550	0.971	0.038	0.986	0.904	0.997
	sRMMD	0.720	0.047	0.741	0.640	0.752	0.998	0.002	0.998	0.994	1.000
	DeepDRK	0.116	0.040	0.100	0.086	0.187	0.791	0.039	0.804	0.720	0.824
2000	DDLK	0.725	0.050	0.734	0.667	0.778	1.000	0.000	1.000	1.000	1.000
	Deep Knockoff	0.626	0.092	0.672	0.460	0.694	1.000	0.001	1.000	0.997	1.000
	KnockoffGAN	0.118	0.014	0.114	0.102	0.139	0.973	0.012	0.978	0.953	0.986
	sRMMD	0.658	0.060	0.644	0.583	0.736	1.000	0.001	1.000	0.998	1.000
	DeepDRK	0.081	0.018	0.076	0.059	0.110	0.973	0.011	0.978	0.956	0.983

Table 1: Comparison of different methods on FDR and power across different (π_1, π_2, π_3) for the components in the Gaussian mixture setup.

Compared to the previous works [55, 38], which consider $\frac{p}{\sqrt{n}}$ as the scaling factor for the Rademacher distribution, we reduce the magnitude of β by a factor of 15. This is because we find that in the original setup, the β scale is too large such that the feature selection enjoys high powers and low FDRs for all models. To compare the performance of the knockoff generation methods on various data, we consider the following distributions for X :

Gaussian mixture: We consider a Gaussian mixture model $X \sim \sum_{k=1}^3 \pi_k \mathcal{N}(\mu_k, \Sigma_k)$, where π is the proportion of the k -th Gaussian with $(\pi_1, \pi_2, \pi_3) = (0.4, 0.2, 0.4)$. $\mu_k \in \mathbb{R}^p$ denotes the mean of the k -th Gaussian with $\mu_k = \mathbf{1}_p \cdot 20 \cdot (k-1)$, where $\mathbf{1}_p$ is the p -dimensional vector that has universal value 1 for all entries. $\Sigma_k \in \mathbb{R}^{p \times p}$ is the covariance matrices whose (i, j) -th entry taking the value $\rho_k^{|i-j|}$, where $\rho_k = \rho_{\text{base}}^{k-0.1}$ and $\rho_{\text{base}} = 0.6$. Besides this experiment, we further perform two additional tests. The first one focuses on a mixture of Gaussians data of various ρ_{base} to study the feature selection performance with highly correlated features. Namely, we consider additional $\rho_{\text{base}} \in \{0.7, 0.8\}$. The second one explores the effect of (π_1, π_2, π_3) to the feature selection performance. In this case, we uniformly draw 10 sets of (π_1, π_2, π_3) , and evaluate the FS performance of all the models considered in this paper. The values of the mixture weights are presented in Table 4 in Appendix H.3.1.

Copulas [49]: We further use copula to model complex correlations within X . To the best of our knowledge, this is a first attempt to consider complex distributions other than the Gaussian mixture model in the knockoff framework. Specifically, we consider two copula families: Clayton, Joe with the consistent copula parameter of 2 in both cases. For each family, we consider two candidates for the marginal distributions: a uniform distribution (using the identity conversion function) and an exponential distribution with a rate of 1. We implement the copulas according to PyCop⁶.

We consider the following (n, p) setups: (200, 100) and (2000, 100). This is in contrast to existing works, which consider only the (2000, 100) as the smallest sample size setup [55, 38, 47]. Our goal is to demonstrate the consistent performance of DeepDRK across various p/n ratios, particularly when the sample size is small.

Results: Figure 2 compare FDRs and powers for all models across the datasets with two different setups for β . Figure 2 shows that DeepDRK consistently controls the false discovery rate (FDR) compared to other benchmark models across various data distributions and p/n ratios, with the exception of a few cases where the FDR exceeds the 0.1 threshold in the small sample size ($n = 200$) scenarios. Other models, though being able to reach higher power, comes at a cost of sacrificing FDR, which contradicts to the UMP philosophy (see Section 3.2). We also evaluate the performance on a mixture of Gaussians with increasing ρ_{base} , indicating a higher correlation among the input variables. Note that the mixture of Gaussians example in Figure 2 has $\rho_{\text{base}} = 0.6$. The results for $\rho_{\text{base}} \in \{0.7, 0.8\}$ are presented in Figure 3. Compared to other baseline models, DeepDRK maintains the lowest FDRs while achieving competitive powers across all ρ_{base} values, highlighting its robustness to correlations in X . Overall, the results demonstrate the ability of DeepDRK in consistently performing FS with controlled FDR compared to other models across a range of different datasets, p/n ratios, and feature correlations in X . In addition, we present common statistics for the FDR and power on the mixture of Gaussians experiment with 10 different sets of (π_1, π_2, π_3) in Table 1. It is clear that DeepDRK improves the robustness in maintaining FDR across various configurations of the Gaussian mixture models compared to existing approaches.

⁶<https://github.com/maximenc/pycop/>

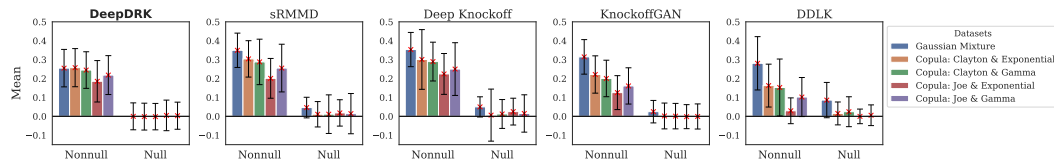


Figure 4: The knockoff statistics (w_j) for different knockoff models on the synthetic datasets with $\beta \sim \frac{p}{15 \cdot \sqrt{N}} \cdot \text{Rademacher}(0.5)$. Each bar in the plot represents the mean of the null/nonnull knockoff statistics averaging on 600 experiments. The error bar indicates the standard deviation. The sample size is 200.

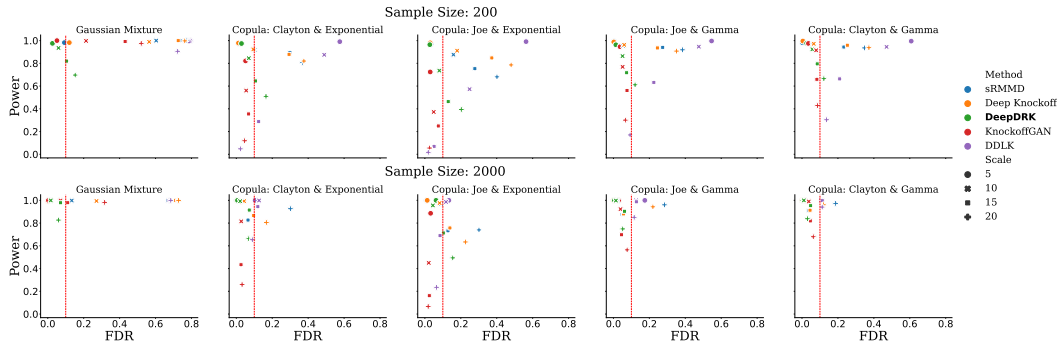


Figure 5: Scatter plots of Power against FDR for different datasets and models. The red vertical line indicates the 0.1 FDR threshold. Different scales for β (e.g., $\frac{p}{5 \cdot \sqrt{n}}$, $\frac{p}{10 \cdot \sqrt{n}}$, $\frac{p}{15 \cdot \sqrt{n}}$ and $\frac{p}{20 \cdot \sqrt{n}}$) are indicated by different marker styles. Different models are indicated by different colors.

To understand why DeepDRK outperforms other baseline models, we consider measuring the distribution of the knockoff statistics, i.e., w_j , for both nonnull and null features of X . Fan et al. [17] and Candès et al. [11] pointed out that a good knockoff requires the corresponding knockoff statistics to concentrate symmetrically around zero for the null features and to maintain high positive values for the nonnulls. However, theoretical analysis on the goodness of FDR or power requires access to the true knockoff $\tilde{X}$ [17] to compare the distribution of w_j 's with the ground truth, which is infeasible for non-Gaussian data. Nevertheless, we can still examine the distribution of the knockoff statistics as a surrogate to analyze model performance in terms of false discovery rate (FDR) or power, given the necessary properties of the knockoff statistics mentioned earlier.

Figure 4 shows the means and standard deviations of the empirical distributions of the knockoff statistics w_j for both null and nonnull variables across different datasets and models. Clearly, compared to other benchmarks, DeepDRK maintains high values of w_j for the nonnulls and relatively symmetric values around zero for the nulls. All other models experience positive shifts to the null statistics to some extent. Positive shifts in the null statistics lead to a degeneracy in performance because the threshold selection rule for false discovery rate (FDR) is based on the negative values of w_j 's (see Eq. (3)). This has two negative impacts, one on the FS threshold and the other on its subsequent FS process. First, according to Eq. (3), the shift causes the chosen threshold to approach zero, as there are fewer null statistics remaining on the negative side, and those that do remain have smaller amplitudes. Subsequently, a lowered threshold leads to an increase in false positives, a phenomenon that becomes more pronounced with positive shifts in the null statistics. As shown in Figure 4, the w_j values calculated using DeepDRK are centered around zero for the nulls, while exhibiting large positive values for the nonnulls. This aligns with the results in Figure 2, which demonstrate that DeepDRK effectively achieves good FDR control and high power. Due to space limit, we defer the results for $n = 2000$ and the comparison of w_j statistics for the Gaussian correlation setup to Appendix I.3.

To further verify the performance consistency of the proposed method, we include a comparison on different β scales. Specifically, we consider 4 different sets of β scales, i.e., $\frac{p}{5 \cdot \sqrt{n}}$, $\frac{p}{10 \cdot \sqrt{n}}$, $\frac{p}{15 \cdot \sqrt{n}}$ and $\frac{p}{20 \cdot \sqrt{n}}$, for the previously considered data distributions. The results are summarized in Figure 5. It is observed that in all cases considered with various β scale, the proposed DeepDRK successfully

maintains relatively low FDRs with a higher power, compared to baseline methods. This phenomenon is especially pronounced with a low sample size ($n = 200$).

In addition to the above results, we provide the measurement of the swap property in Appendix I.1. The evaluation of model runtime is also included in Appendix I.4.

4.2 The Semi-synthetic Experiments

We consider a semi-synthetic study with design X drawn from two real-world datasets and use X to simulate response Y . The first dataset contains single-cell RNA sequencing (scRNA-seq) data from 10× Genomics⁷. Each entry in $X \in \mathbb{R}^{n \times p}$ represents the observed gene expression of p genes in n cells. We refer readers to [23] and [1] for more background. Following the same preprocessing in [23], we obtain the final dataset X with $n = 10000$ and $p = 100$ ⁸. The preprocessing of X and the synthesis of Y (denoted as “Linear” and “Tanh” cases) are deferred to Appendix H.3.2.

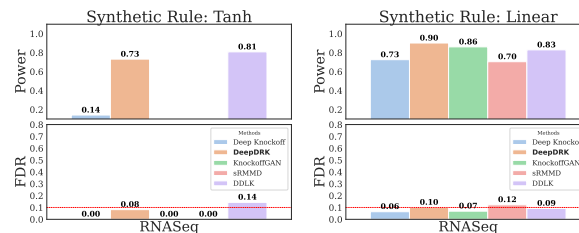


Figure 6: Power and FDR for different knockoff models on the semi-synthetic RNA dataset. The red horizontal line indicates the 0.1 FDR threshold.

The second publicly available dataset⁹ is from a real case study entitled “Longitudinal Metabolomics of the Human Microbiome in Inflammatory Bowel Disease (IBD)” [35]. The study seeks to identify important metabolites of two representative diseases of the inflammatory bowel disease (IBD): ulcerative colitis (UC) and Crohn’s disease (CD). Specifically, we use the C18 Reverse-Phase Negative Mode dataset that has 546 samples and 91 metabolites. To mitigate the effects of missing values, we preprocess the dataset following a common procedure to remove metabolites that have over 20% missing values, resulting in 80 metabolites. We normalize the data matrix entry-wise to have zero mean and unit variance after a log transform and an imputation via the k -nearest neighbor algorithm following the same procedure in [38]. Finally, we synthesize the response Y with the real dataset of X via $Y \sim \mathcal{N}(X^T \beta, 1)$, where the entries of β drawn independently from one of the following three distributions: $\text{Unif}(0, 1)$, $\mathcal{N}(0, 1)$, and $\text{Rademacher}(0.5)$, in three separate experiments.

Results: Figure 6 and 7 compare the feature selection performance on the RNA data and the IBD data respectively. In Figure 6, we observe that all but DDLK are bounded by the nominal 0.1 FDR threshold in the “Tanh” case. However, KnockoffGAN and sRMMD have almost zero power. The power for Deep Knockoff is also very low compared to that of DeepDRK. Although DDLK provides high power, the associated FDR is not controlled by the threshold. In the “Linear” case, almost all models have well controlled FDR, among which DeepDRK provides the highest power. Similar observations can be found in Figure 7. For the IBD data generated under the aforementioned synthesis rules, it is clear that all models except DDLK achieve well-controlled FDR. Apart from DDLK, DeepDRK consistently demonstrates the highest power. These results further underscore the potential of DeepDRK for real-world data applications.

4.3 A Case Study

Besides (semi-)synthetic setups, we carry out a case study with real data for both design X and response Y , in order to qualitatively evaluate the selection performance of DeepDRK. In this subsection, we use the IBD dataset [35] with the empirical response. The response variable Y is categorical: Y equals 1 if a given sample is associated with UC/CD and 0 otherwise. The covariates X is identical to the second semi-synthetic setup considered in Section 4.2. To properly evaluate results with no ground truth available, we search for the evidence of the IBD-associated metabolites using the existing literature. Specifically, we use three sources: 1. metabolites that are explicitly

⁷<https://kb.10xgenomics.com/hc/en-us>

⁸The data processing code is adopted from this repo: <https://github.com/dereklhansen/flowselect/tree/master/data>

⁹<https://www.metabolomicsworkbench.org/> under the project DOI: 10.21228/M82T15.

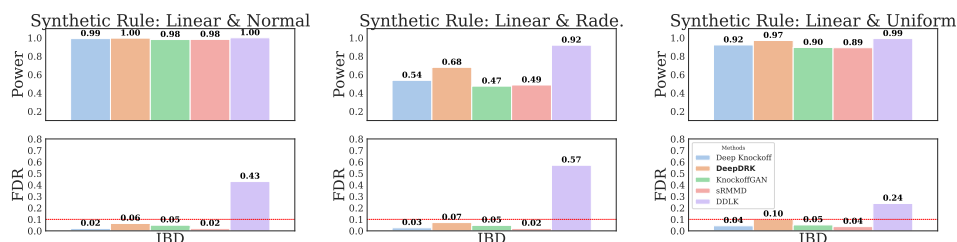


Figure 7: Power and FDR for different knockoff models on the semi-synthetic IBD dataset. The red horizontal line indicates the 0.1 FDR threshold.

Model	DeepDRK	Deep Knockoff	sRMMD	KnockoffGAN	DDLK
Referenced / Identified	19/23	15/20	5/5	12/14	17/25

Table 2: The number of literature-supported metabolites among the identified metabolites vs. the number of identified metabolites.

documented to have associations with IBD, UC, or CD in the PubChem database ¹⁰; 2. metabolites that are reported in the existing peer-reviewed publications; 3. metabolites that are reported in pre-prints. We identify 47 metabolites that are reported to have association with IBD. All referenced metabolites are included in Table 5 in Appendix H.3.3.

Our DeepDRK model training and knockoff generation are the same as before (see Table 3 in Appendix H.1). Likewise, to generate knockoff for the benchmark models, we follow their default setups. During the FS step, however, we use 0.2 as the FDR threshold instead of 0.1, and apply a different algorithm—DeepPINK [37] that is included in the knockpy ¹¹ library—to generate w_j . The values are subsequently used to identify metabolites. We choose DeepPINK over the previously considered ridge regression due to the nonlinear relationships between metabolites X and responses Y in this case study.

We compare the FS results with the 47 literature-supported metabolites and report the number of selections in Table 2. A detailed list of selected features for each model can be found in Table 9 in Appendix I.5. From Table 2, it is clear that, compared to the benchmark models, DeepDRK identifies the largest number of referenced metabolites while effectively limiting the number of metabolites not reported in existing literature (see Table 5 in Appendix H.3.3). The sRMMD model achieves the lowest false discovery rate, but this comes at the cost of missing a significant number of documented metabolites. Since there is no ground truth available, the results here should be viewed and analyzed qualitatively.

5 Conclusion

In this paper, we introduce DeepDRK, a deep learning-based knockoff generation pipeline consisting of two steps. First, it trains a Knockoff Transformer with multiple swappers to achieve the swap property while reducing reconstructability. In the post-training stage, a dependency-regularized perturbation is applied to further enhance power with controlled FDR. DeepDRK effectively balances FDR and power, which compete with each other at the sample level. To the best of our knowledge, this relationship has not been previously reported in the literature. Empirically, DeepDRK demonstrates the ability to maintain both controlled false discovery rates (FDR) and high power across various data distributions and different p/n ratios. Additionally, we provide insights into the distribution of knockoff statistics, which elucidate the reasons behind DeepDRK's consistently strong performance. The numerical results indicate that DeepDRK outperforms existing deep learning-based benchmark models. Experiments with real and semi-synthetic data further highlight the potential of DeepDRK for feature selection tasks involving non-Gaussian data.

¹⁰<https://pubchem.ncbi.nlm.nih.gov/>

¹¹<https://amspector100.github.io/knockpy/>

Acknowledgments

This research was partially supported by Alfred P. Sloan foundation and NSF #1934757.

References

- [1] Agarwal, D., Wang, J., and Zhang, N. R. Data denoising and post-denoising corrections in single cell RNA sequencing. *Statistical Science*, 35(1):112 – 128, 2020.
- [2] Ananthakrishnan, A. N., Luo, C., Yajnik, V., Khalili, H., Garber, J. J., Stevens, B. W., Cleland, T., and Xavier, R. J. Gut microbiome function predicts response to anti-integrin biologic therapy in inflammatory bowel diseases. *Cell host & microbe*, 21(5):603–610, 2017.
- [3] Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *ICML*, pp. 214–223, 2017.
- [4] Barber, R. F. and Candès, E. J. Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, 43(5):2055–2085, 2015.
- [5] Bauset, C., Gisbert-Ferrándiz, L., and Cosín-Roger, J. Metabolomics as a promising resource identifying potential biomarkers for inflammatory bowel disease. *Journal of Clinical Medicine*, 10(4):622, 2021.
- [6] Benjamini, Y. and Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.
- [7] Bin Masud, S., Jenkins, C., Hussey, E., Elkin-Frankston, S., Mach, P., Dhummakupt, E., and Aeron, S. Utilizing machine learning with knockoff filtering to extract significant metabolites in Crohn’s disease with a publicly available untargeted metabolomics dataset. *Plos one*, 16(7): e0255240, 2021.
- [8] Blaker, P. A., Arenas-Hernandez, M., Smith, M. A., Shobowale-Bakre, E. A., Fairbanks, L., Irving, P. M., Sanderson, J. D., and Marinaki, A. M. Mechanism of allopurinol induced TPMT inhibition. *Biochemical pharmacology*, 86(4):539–547, 2013.
- [9] Bonneel, N., Rabin, J., Peyré, G., and Pfister, H. Sliced and Radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51:22–45, 2015.
- [10] Bonnotte, N. *Unidimensional and evolution methods for optimal transportation*. PhD thesis, Université Paris Sud-Paris XI; Scuola normale superiore (Pise, Italie), 2013.
- [11] Candès, E., Fan, Y., Janson, L., and Lv, J. Panning for gold: ‘Model-X’ knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577, 2018.
- [12] Casella, G. and Berger, R. L. *Statistical inference*. Cengage Learning, 2021.
- [13] Deshpande, I., Zhang, Z., and Schwing, A. G. Generative modeling using the sliced Wasserstein distance. In *CVPR*, pp. 3483–3491, 2018.
- [14] Deshpande, I., Hu, Y.-T., Sun, R., Pyrros, A., Siddiqui, N., Koyejo, S., Zhao, Z., Forsyth, D., and Schwing, A. G. Max-sliced Wasserstein distance and its use for GANs. In *CVPR*, pp. 10648–10656, 2019.
- [15] Dhal, P. and Azad, C. A comprehensive survey on feature selection in the various fields of machine learning. *Applied Intelligence*, 52(4):4543–4581, 2022.
- [16] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16×16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- [17] Fan, Y., Gao, L., and Lv, J. ARK: Robust Knockoffs inference with coupling. *arXiv preprint arXiv:2307.04400*, 2023.

- [18] Fretland, D., Widomski, D., Levin, S., and Gaginella, T. Colonic inflammation in the rabbit induced by phorbol-12-myristate-13-acetate. *Inflammation*, 14(2):143–150, 1990.
- [19] Gavrilov, Y., Benjamini, Y., and Sarkar, S. K. An adaptive step-down procedure with proven FDR control under independence. *The Annals of Statistics*, 37(2):619–629, 2009.
- [20] Gimenez, J. R., Ghorbani, A., and Zou, J. Knockoffs for the mass: new feature importance statistics with false discovery guarantees. In *AISTATS*, pp. 2125–2133, 2019.
- [21] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- [22] Guyon, I. and Elisseeff, A. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.
- [23] Hansen, D., Manzo, B., and Regier, J. Normalizing flows for knockoff-free controlled feature selection. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16125–16137, 2022.
- [24] He, Z., Liu, L., Wang, C., Le Guen, Y., Lee, J., Gogarten, S., Lu, F., Montgomery, S., Tang, H., Silverman, E. K., et al. Identification of putative causal loci in whole-genome sequencing data via knockoff statistics. *Nature communications*, 12(1):1–18, 2021.
- [25] Hristache, M., Juditsky, A., and Spokoiny, V. Direct estimation of the index coefficient in a single-index model. *Annals of Statistics*, 29(3):595–623, 2001.
- [26] Jang, E., Gu, S., and Poole, B. Categorical reparameterization with Gumbel-softmax. In *ICLR*, 2017.
- [27] Jordon, J., Yoon, J., and van der Schaar, M. KnockoffGAN: Generating knockoffs for feature selection using generative adversarial networks. In *ICLR*, 2018.
- [28] Kolouri, S., Nadjahi, K., Simsekli, U., Badeau, R., and Rohde, G. Generalized sliced Wasserstein distances. In *NeurIPS*, volume 32, 2019.
- [29] Koon, H. W. A novel orally active metabolite reverses Crohn’s disease-associated intestinal fibrosis. *Inflammatory Bowel Diseases*, 28(Supplement_1):S61–S62, 2022.
- [30] Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., and Courville, A. Out-of-distribution generalization via risk extrapolation (REx). In *ICML*, pp. 5815–5826, 2021.
- [31] Lavelle, A. and Sokol, H. Gut microbiota-derived metabolites as key actors in inflammatory bowel disease. *Nature reviews Gastroenterology & hepatology*, 17(4):223–237, 2020.
- [32] Lee, T., Clavel, T., Smirnov, K., Schmidt, A., Lagkouvardos, I., Walker, A., Lucio, M., Michalke, B., Schmitt-Kopplin, P., Fedorak, R., et al. Oral versus intravenous iron replacement therapy distinctly alters the gut microbiota and metabolome in patients with IBD. *Gut*, 66(5):863–871, 2017.
- [33] Li, T., Yu, J., and Meng, C. Scalable model-free feature screening via sliced-Wasserstein dependency. *Journal of Computational and Graphical Statistics*, 32(4):1501–1511, 2023.
- [34] Liu, Y. and Zheng, C. Auto-encoding knockoff generator for FDR controlled variable selection. *arXiv preprint arXiv:1809.10765*, 2018.
- [35] Lloyd-Price, J., Arze, C., Ananthakrishnan, A. N., Schirmer, M., Avila-Pacheco, J., Poon, T. W., Andrews, E., Ajami, N. J., Bonham, K. S., Brislawn, C. J., et al. Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases. *Nature*, 569(7758):655–662, 2019.
- [36] Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *ICLR*, 2019.
- [37] Lu, Y., Fan, Y., Lv, J., and Stafford Noble, W. DeepPINK: reproducible feature selection in deep neural networks. In *NeurIPS*, volume 31, 2018.

- [38] Masud, S. B., Werenski, M., Murphy, J. M., and Aeron, S. Multivariate soft rank via entropy-regularized optimal transport: Sample efficiency and generative modeling. *Journal of Machine Learning Research*, 24(160):1–65, 2023.
- [39] Mehta, R. S., Taylor, Z. L., Martin, L. J., Rosen, M. J., and Ramsey, L. B. *SLCO1B1* *15 allele is associated with methotrexate-induced nausea in pediatric patients with inflammatory bowel disease. *Clinical and translational science*, 15(1):63–69, 2022.
- [40] Minderhoud, I. M., Oldenburg, B., Schipper, M. E. I., Ter Linde, J. J. M., and Samsom, M. Serotonin synthesis and uptake in symptomatic patients with Crohn’s disease in remission. *Clinical Gastroenterology and Hepatology*, 5(6):714–720, 2007.
- [41] Nadjahi, K., Durmus, A., Simsekli, U., and Badeau, R. Asymptotic guarantees for learning generative models with the sliced-Wasserstein distance. In *Advances in Neural Information Processing Systems*, volume 32, 2019.
- [42] Narasimhan, R. L., Throm, A. A., Koshy, J. J., Saldanha, K. M. R., Chandranpillai, H., Lal, R. D., Kumravat, M., K. M., A. K., Batra, A., Zhong, F., et al. Inferring intestinal mucosal immune cell associated microbiome species and microbiota-derived metabolites in inflammatory bowel disease. *bioRxiv*, 2020.
- [43] Nies, T. G., Staudt, T., and Munk, A. Transport dependency: Optimal transport based dependency measures. *arXiv preprint arXiv:2105.02073*, 2021.
- [44] Nuzzo, A., Saha, S., Berg, E., Jayawickreme, C., Tocker, J., and Brown, J. R. Expanding the drug discovery space with predicted metabolite–target interactions. *Communications biology*, 4(1):1–11, 2021.
- [45] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, volume 32, 2019.
- [46] Qin, X. Etiology of inflammatory bowel disease: a unified hypothesis. *World journal of gastroenterology: WJG*, 18(15):1708, 2012.
- [47] Romano, Y., Sesia, M., and Candès, E. Deep knockoffs. *Journal of the American Statistical Association*, 115(532):1861–1872, 2020.
- [48] Saber, S., Khalil, R. M., Abdo, W. S., Nassif, D., and El-Ahwany, E. Olmesartan ameliorates chemically-induced ulcerative colitis in rats via modulating NF κ B and Nrf-2/HO-1 signaling crosstalk. *Toxicology and applied pharmacology*, 364:120–132, 2019.
- [49] Schmidt, T. Coping with copulas. *Copulas-From theory to application in finance*, 3:1–34, 2007.
- [50] Scoville, E. A., Allaman, M. M., Brown, C. T., Motley, A. K., Horst, S. N., Williams, C. S., Koyama, T., Zhao, Z., Adams, D. W., Beaulieu, D. B., et al. Alterations in lipid, amino acid, and energy metabolism distinguish Crohn’s disease from ulcerative colitis and control subjects by serum metabolomic profiling. *Metabolomics*, 14(1):1–12, 2018.
- [51] Sesia, M., Sabatti, C., and Candès, E. J. Gene hunting with hidden Markov model knockoffs. *Biometrika*, 106(1):1–18, 08 2018.
- [52] Soderholm, J. D., Oman, H., Blomquist, L., Veen, J., Lindmark, T., and Olaison, G. Reversible increase in tight junction permeability to macromolecules in rat ileal mucosa in vitro by sodium caprate, a constituent of milk fat. *Digestive diseases and sciences*, 43(7):1547–1552, 1998.
- [53] Söderholm, J. D., Olaison, G., Peterson, K. H., Franzen, L. E., Lindmark, T., Wirén, M., Tagesson, C., and Sjö Dahl, R. Augmented increase in tight junction permeability by luminal stimuli in the non-inflamed ileum of Crohn’s disease. *Gut*, 50(3):307–313, 2002.
- [54] Spector, A. and Janson, L. Powerful knockoffs via minimizing reconstructability. *The Annals of Statistics*, 50(1):252–276, 2022.
- [55] Sudarshan, M., Tansey, W., and Ranganath, R. Deep direct likelihood knockoffs. In *NeurIPS*, volume 33, 2020.

- [56] Suhre, K., Shin, S.-Y., Petersen, A.-K., Mohny, R. P., Meredith, D., Wägele, B., Altmaier, E., Deloukas, P., Erdmann, J., Grundberg, E., et al. Human metabolic individuality in biomedical and pharmaceutical research. *Nature*, 477(7362):54–60, 2011.
- [57] Uchiyama, K., Odahara, S., Nakamura, M., Koido, S., Katahira, K., Shiraishi, H., Ohkusa, T., Fujise, K., and Tajiri, H. The fatty acid profile of the erythrocyte membrane in initial-onset inflammatory bowel disease patients. *Digestive diseases and sciences*, 58(5):1235–1243, 2013.
- [58] Uko, V., Thangada, S., and Radhakrishnan, K. Liver disorders in inflammatory bowel disease. *Gastroenterology research and practice*, 2012(1):642923, 2012.
- [59] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, volume 30, 2017.
- [60] Villani, C. *Optimal transport: old and new*, volume 338. Springer, 2009.
- [61] Wiesel, J. C. Measuring association with Wasserstein distances. *Bernoulli*, 28(4):2816–2832, 2022.
- [62] Zhu, Z., Fan, Y., Kong, Y., Lv, J., and Sun, F. DeepLINK: Deep learning inference using knockoffs with applications to genomics. *Proceedings of the National Academy of Sciences*, 118(36):e2104683118, 2021.

Appendix

A Reconstructability and Selection Power

The concept of reconstructability was introduced by [54] as a population-level counterpart to what is commonly referred to as collinearity in linear regression. Under a Gaussian design where $X \sim \mathcal{N}(0, \Sigma)$, reconstructability is high if Σ is not of full rank, implying that there exists some j such that X_j is almost surely a linear combination of X_{-j} . More generally, if there exists more than one representation of the response Y using the explanatory variable X , we qualitatively consider the reconstructability to be high. High collinearity often impairs statistical power, and similarly, high reconstructability diminishes feature selection power. To illustrate this, we state a linear version of Theorem 2.3 from [54], originally formulated for a more general single-index model [25].

Theorem A.1. *Let $Y = X_J \beta_J + X_{-J} \beta_{-J} + \varepsilon$, where $J \subset [p]$ and ε is centered Gaussian noise. Equivalently, this implies $Y \perp\!\!\!\perp X_J \mid X_J \beta_J, X_{-J}$. Suppose there exists a β_J^* such that $X_J \beta_J = X_J \beta_J^*$ almost surely. Then, denoting $Y^* = X_J \beta_J^* + X_{-J} \beta_{-J} + \varepsilon$, we have*

$$\begin{aligned} ((X, \tilde{X}), Y) &\stackrel{d}{=} ((X, \tilde{X})_{\text{swap}(J)}, Y^*) \quad \text{and} \\ ((X, \tilde{X}), Y^*) &\stackrel{d}{=} ((X, \tilde{X})_{\text{swap}(J)}, Y). \end{aligned} \quad (10)$$

Furthermore, in the knockoff framework [4], let $w = w((X, \tilde{X}), y)$ and $w^* = w((X, \tilde{X}), y^*)$. Then, for all $j \in J$,

$$\mathbb{P}(w_j > 0) + \mathbb{P}(w_j^* > 0) \leq 1. \quad (11)$$

Eq. (11) implies a “no free lunch” situation for selection power when there is exact reconstructability.

To address the reconstructability issue, [54] proposed two methods in Gaussian design. The first is the minimal variance-based reconstructability (MVR) knockoff, in which the knockoff $\tilde{X}$ is sampled to minimize the loss

$$L_{\text{MVR}} = \sum_{j=1}^p \frac{1}{\mathbb{E} \left[\text{Var} \left(X_j \mid X_{-j}, \tilde{X} \right) \right]}. \quad (12)$$

Note that this is equivalent to maximizing $\mathbb{E} \left[\text{Var} \left(X_j \mid X_{-j}, \tilde{X} \right) \right]$ for all $j \in [p]$. Another approach is the maximum entropy (ME) knockoff, where $\tilde{X}$ is sampled to maximize

$$L_{\text{ME}} = \int \int p(x, \tilde{x}) \log(p(x, \tilde{x})) d\tilde{x} dx. \quad (13)$$

Under Gaussian design, both optimizations have closed-form solutions. Since X is Gaussian, $(X, \tilde{X})$ must also be jointly Gaussian to satisfy the swap property. To optimize, one first calculates the covariance matrix using the SDP method in [4], yielding a diagonal matrix S . Then, both MVR and ME reduce to an optimization on S :

$$\begin{aligned} L_{\text{MVR}}(S) &\propto \text{Tr} \left(G_S^{-1} \right) = \sum_{j=1}^{2p} \frac{1}{\lambda_j(G_S)} \\ \text{and} \quad L_{\text{ME}}(S) &= \log \det \left(G_S^{-1} \right) = \sum_{j=1}^{2p} \log \left(\frac{1}{\lambda_j(G_S)} \right). \end{aligned} \quad (14)$$

Although both methods show high power for feature selection, neither MVR nor ME can be directly extended to arbitrary distributions due to the intractability of conditional variance and likelihood.

In DeepDRK, we consider regularizing with a sliced-Wasserstein-based dependency correlation, which can be considered a stronger dependency regularization than entropy. A post-training perturbation is also applied to further reduce collinearity. However, the theoretical understanding of how these affect the swap property and power remains an open question.

B DeepDRK Model Architecture and Training Algorithm

B.1 Model Architecture

DeepDRK's knockoff Transformer (KT) model is based on the popular Vision Transformer (ViT) [16]. The primary difference is that the input dimension is 1D, not 2D, for X . We do not use patches as input; instead, we treat all entries of X to account for correlations between each pair of entries in the knockoff $\tilde{X}$. This structure is similar to the original Transformer [59]. Nonetheless, we retain other components from ViT, such as patch embedding, PreNorm, and 1D-positional encoding [16]. Since knockoff generation requires a distribution, we feed X and a uniformly distributed random variable Z with the same dimension as X to inject randomness. Specifically for the DeepDRK model, we use 8 attention heads, 8 layers, and a hidden dimension of 512 in the ViT model.

The swapper module, first introduced in DDLK [55], produces the index subset B for the adversarial swap attack. Optimizing knockoff against these adversarial swaps enforces the swap property. Specifically, the swapper consists of a matrix with shape $2 \times p$ (i.e., trainable model weights), where p is the dimension of X . This matrix controls the Gumbel-softmax distribution [26] for all p entries. Each entry is represented by a binary Gumbel-softmax random variable (i.e., it can only take values of 0 or 1). To generate the subset B , we sample b_j from the corresponding j -th Gumbel-softmax random variable, and B is defined as $\{j \in [p] ; b_j = 1\}$. During optimization, we maximize Eq. (5) with respect to the weights ω_i of the swapper S_{ω_i} , so that the sampled indices, with which the swap is applied, lead to a higher SWD in the objective in Eq. (5). Minimizing this objective with respect to $\tilde{X}_\theta$ requires the knockoff to counteract the adversarial swaps, thereby enforcing the swap property. Compared to DDLK, the proposed DeepDRK utilizes multiple independent swappers (i.e., $K = 2$). And we set the temperature for the Gumbel-softmax to be 0.2.

B.2 Training Algorithm

In Algorithm 1, we provide pseudo code for training the Knockoff Transformer and the swappers (i.e., the first stage shown in Figure 1).

Algorithm 1 DeepDRK Training

```

1: Input: Knockoff transformer  $\tilde{X}_\theta$ , denoted as  $g_\theta(\cdot)$ ; swappers  $S_\omega$ ; number of swappers  $K$ ;
   learning rate  $\alpha_s$  for the swappers; learning rate  $\alpha_\theta$  for the knockoff transformer; early stop
   tolerance  $\eta$ ; number of epochs  $T$ ; batch size  $B_s$ ; dataset  $\mathcal{D}$ ; swapper update frequency  $\gamma = 3$ 
2: Output:  $\theta$  for  $\tilde{X}_\theta$ 
3: Split dataset  $\mathcal{D}$  into training set  $\mathcal{D}_{\text{train}}$  and validation set  $\mathcal{D}_{\text{val}}$ 
4: Initialize the knockoff transformer  $g_\theta(\cdot)$  with random weights
5: Initialize swappers  $S_{\omega_i}$ ,  $i = 1, \dots, K$  with random weights
6: Initialize the AdamW optimizer  $\text{opt}_\theta$  with learning rate  $\alpha_\theta$  for  $g_\theta(\cdot)$ 
7: Initialize the AdamW optimizer  $\text{opt}_{\omega_i}$  with learning rate  $\alpha_s$  for  $S_{\omega_i}$ ,  $i = 1, \dots, K$ 
8: for  $t = 1$  to  $T$  do
9:   for  $l = 1$  to  $\frac{|\mathcal{D}_{\text{train}}|}{B_s}$  do
10:    Sample  $B_s$  samples of  $X$  from  $\mathcal{D}_{\text{train}}$ :  $X_l$ 
11:    Generate knockoff  $\tilde{X}_l = g_\theta(X_l)$ 
12:    Calculate  $\mathcal{L}_{\text{SL}}(X_l, \tilde{X}_l, \{S_{\omega_i}\}_{i=1}^K)$  and  $\mathcal{L}_{\text{DRL}}(X_l, \tilde{X}_l)$ 
13:     $\theta \leftarrow \theta + \text{opt}_\theta(\mathcal{L}_{\text{SL}}(X_l, \tilde{X}_l, \{S_{\omega_i}\}_{i=1}^K) + \mathcal{L}_{\text{DRL}}(X_l, \tilde{X}_l))$ 
14:    if  $l \bmod \gamma = 0$  then
15:       $\omega_i \leftarrow \omega_i + \text{opt}_{\omega_i}(-\mathcal{L}_{\text{SL}}(X_l, \tilde{X}_l, \{S_{\omega_i}\}_{i=1}^K)), i = 1, \dots, K$ 
16:    end if
17:  end for
18:  Calculate the validation loss on  $\mathcal{D}_{\text{val}}$ :  $\mathcal{L}_{\text{SL}}^{\text{val}} + \mathcal{L}_{\text{DRL}}^{\text{val}}$ 
19:  if  $\mathcal{L}_{\text{SL}}^{\text{val}} + \mathcal{L}_{\text{DRL}}^{\text{val}}$  meets early stop condition at tolerance  $\eta$  then
20:    break
21:  end if
22: end for

```

C From Wasserstein to Sliced Wasserstein Distance

The Wasserstein distance has gained popularity in both mathematics and machine learning due to its ability to compare different types of distributions [60] and its differentiability [3]. Here, we provide its definition. Let X, Y be two $\mathbb{R}^d$ random vectors following distributions P_X, P_Y with finite p -th moments. The *Wasserstein- p* distance between P_X and P_Y is:

$$W_p(P_X, P_Y) = \inf_{\gamma \in \Gamma(P_X, P_Y)} (\mathbb{E}_{(x,y) \sim \gamma} \|x - y\|^p)^{\frac{1}{p}} \quad (15)$$

where $\Gamma(P_X, P_Y)$ denotes the set of all joint distributions such that their marginals are P_X and P_Y . When $d = 1$, the Wasserstein distance between two one-dimensional distributions can be written as:

$$\begin{aligned} W_p(P_X, P_Y) &= \left(\int_0^1 |F_X^{-1}(v) - F_Y^{-1}(v)|^p dv \right)^{\frac{1}{p}} \\ &= \|F_X^{-1} - F_Y^{-1}\|_{L^p([0,1])}, \end{aligned} \quad (16)$$

where F_X and F_Y are the cumulative distribution functions (CDFs) of P_X and P_Y , respectively. Moreover, if $p = 1$, the Wasserstein distance further simplifies to

$$W_1(P_X, P_Y) = \int |F_X(v) - F_Y(v)| dv = \|F_X - F_Y\|_{L^1(\mathbb{R})}. \quad (17)$$

From the above, it is evident that the 1D Wasserstein distance is straightforward to compute, leading to the development of the sliced Wasserstein distance (SWD) [9]. To leverage this computational advantage in 1D, one first projects both distributions uniformly onto a 1D direction and computes the Wasserstein- p distance between the two projected distributions. SWD is then calculated by taking the expectation over the random direction. Specifically, let $\mu \in \mathbb{S}^{d-1}$ denote a projection direction, and the push-forward distribution [60] $\mu_{\#}P_X$ denotes the law of $\mu^T X$. When μ is uniformly distributed on the d -dimensional sphere, the p -sliced Wasserstein distance between P_X and P_Y is given by:

$$SW_p(P_X, P_Y) = \int_{\mu \in \mathbb{S}^{d-1}} W_p(\mu_{\#}P_X, \mu_{\#}P_Y) d\mu. \quad (18)$$

Combining Eq. (18) and Eq. (16) yields:

$$\begin{aligned} SW_p(P_X, P_Y) &= \\ &= \int_{\mu \in \mathbb{S}^{d-1}} \left(\int_0^1 |(F_X^{\mu})^{-1}(v) - (F_Y^{\mu})^{-1}(v)|^p dv \right)^{\frac{1}{p}} d\mu \end{aligned} \quad (19)$$

$$= \int_{\mu \in \mathbb{S}^{d-1}} \int |F_X^{\mu}(v) - F_Y^{\mu}(v)| dv d\mu \quad \text{when } p = 1. \quad (20)$$

Despite faster computation, the convergence of SWD has been shown to be equivalent to the convergence of the Wasserstein distance under mild conditions [10]. In practice, the expectation over μ is approximated by a finite summation over a number of projection directions chosen uniformly from $\mathbb{S}^{d-1}$.

D Sliced Wasserstein Correlation

The idea of metricizing independence is recently advanced using the Wasserstein distance [61, 43]. Given a joint distribution $(X, Y) \sim \Gamma_{XY}$ and its marginal distributions $X \sim P_X, Y \sim P_Y$, the Wasserstein Dependency (WD) between X and Y is defined by $WD(X, Y) = W_p(\Gamma_{XY}, P_X \otimes P_Y)$. A trivial observation is that $WD(X, Y) = 0$ implies that X and Y are independent. Due to the high computational cost of Wasserstein distance, sliced Wasserstein dependency (SWDep) [33] is developed using sliced Wasserstein distance (see Appendix C for SWD details). The SW dependency (SWDep) between X and Y is defined as $SW_p(\Gamma_{XY}, P_X \otimes P_Y)$, and a zero SWDep indicates independence. Since the dependency metric is not bounded from above, sliced Wasserstein correlation

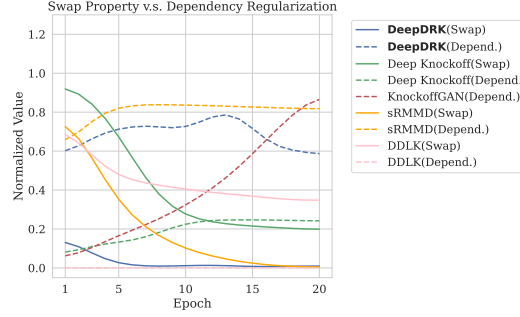


Figure 8: The competing relationship between the swap property ($\mathcal{L}_{\text{SL}}$ in solid curves) and dependency regularization ($\mathcal{L}_{\text{DRL}}$ in dashed curves).

(SWC) is introduced to normalize SWDep. More specifically, the SWC between X and Y is defined as

$$\begin{aligned} \text{SWC}_p(X, Y) &:= \frac{\text{SWDep}_p(X, Y)}{\sqrt{\text{SWDep}_p(X, X) \text{SWDep}_p(Y, Y)}} \\ &= \frac{\text{SWD}_p(\Gamma_X Y, P_X \otimes P_Y)}{\sqrt{\text{SWD}_p(\Gamma_X X, P_X \otimes P_X) \text{SWD}_p(\Gamma_Y Y, P_Y \otimes P_Y)}}, \end{aligned} \quad (21)$$

where $\Gamma_X X$ and $\Gamma_Y Y$ are the joint distributions of (X, X) and (Y, Y) respectively. It is shown that $0 \leq \text{SWC}_p(X, Y) \leq 1$ and $\text{SWC}_p(X, Y) = 1$ when X has a linear relationship with Y [33].

In terms of computing SWC, we follow [33] and consider both laws of X and Y to be sums of $2n$ Diracs, i.e., both variables are empirical distributions with data $\mathcal{I}_{\text{full}} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{2n}$. Define $\mathcal{I} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ and $\tilde{\mathcal{I}} = \{(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^n$, where $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i) = (\mathbf{x}_{n+i}, \mathbf{y}_{n+i})$. Because data is i.i.d., $\mathcal{I}$ and $\tilde{\mathcal{I}}$ are independent. We further introduce the following notation:

$$\begin{aligned} \mathcal{I}_{\mathbf{xy}} &= \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n, \tilde{\mathcal{I}}_{\mathbf{xy}} = \{(\tilde{\mathbf{x}}_i, \mathbf{y}_i)\}_{i=1}^n \\ \mathcal{I}_{\mathbf{xx}} &= \{(\mathbf{x}_i, \mathbf{x}_i)\}_{i=1}^n, \tilde{\mathcal{I}}_{\mathbf{xx}} = \{(\tilde{\mathbf{x}}_i, \mathbf{x}_i)\}_{i=1}^n, \\ \mathcal{I}_{\mathbf{yy}} &= \{(\mathbf{y}_i, \mathbf{y}_i)\}_{i=1}^n, \tilde{\mathcal{I}}_{\mathbf{yy}} = \{(\tilde{\mathbf{y}}_i, \mathbf{y}_i)\}_{i=1}^n \end{aligned}$$

Then the empirical SWC can be computed by:

$$\widehat{\text{SWC}}_p(X, Y) := \frac{\text{SWD}_p(\mathcal{I}_{\mathbf{xy}}, \tilde{\mathcal{I}}_{\mathbf{xy}})}{\sqrt{\text{SWD}_p(\mathcal{I}_{\mathbf{xx}}, \tilde{\mathcal{I}}_{\mathbf{xx}}) \text{SWD}_p(\mathcal{I}_{\mathbf{yy}}, \tilde{\mathcal{I}}_{\mathbf{yy}})}}. \quad (22)$$

E Competing Losses

In Figure 8, we present scaled $\mathcal{L}_{\text{SL}}$ and $\mathcal{L}_{\text{DRL}}$ curves for each model considered. For better visualization, these curves are normalized to values between 0 and 1. We observe the first 20 epochs, as $\mathcal{L}_{\text{DRL}}$ flattens out in later epochs without further decrease. The competition between losses is evident: as $\mathcal{L}_{\text{SL}}$ decreases, $\mathcal{L}_{\text{DRL}}$ increases, indicating difficulty in maintaining low reconstructability.

F Proofs

F.1 Proof of Lemma 3.1

Assuming all conditions in [41] hold, we prove this by applying inequalities on $\text{SWD}(\hat{P}_n, \hat{P}_n^B)$ and $\text{SWD}(P, P^B)$:

$$\begin{aligned} \text{SWD}(\hat{P}_n, \hat{P}_n^B) &\leq \text{SWD}(\hat{P}_n, P) + \text{SWD}(P, P^B) + \text{SWD}(P^B, \hat{P}_n^B) \\ &= \text{SWD}(P, P^B) + \mathcal{O}(n^{-1/2}), \end{aligned} \quad (23)$$

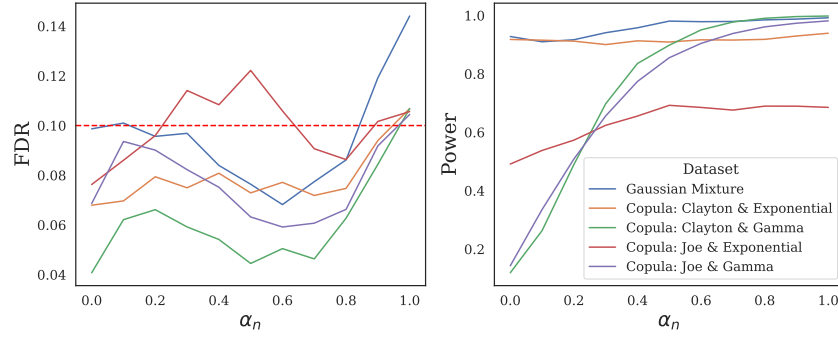


Figure 9: The effect of α_n for $\tilde{X}_{\theta}^{\text{DRP}}$ in Eq. (8) on FDR and power. When α_n is 0, we consider the knockoff generated from the knockoff transformer without any dependency regularization perturbation. When α_n is 1, there is only perturbation X_{rp} without the knockoff. The sample size $n = 2000$.

where the first inequality is a direct application of the triangle inequality, and the second equality follows from Theorem 5 in [41]. Similarly, we obtain

$$\begin{aligned} \text{SWD}(P, P^B) &\leq \text{SWD}(P, \hat{P}_n) + \text{SWD}(\hat{P}_n, \hat{P}_n^B) + \text{SWD}(\hat{P}_n^B, P^B) \\ &= \text{SWD}(\hat{P}_n, \hat{P}_n^B) + \mathcal{O}(n^{-1/2}). \end{aligned} \quad (24)$$

Combining Eq. (23) and Eq. (24), we have

$$\text{SWD}(P, P^B) = \text{SWD}(\hat{P}_n, \hat{P}_n^B) + \mathcal{O}(n^{-1/2}). \quad (25)$$

F.2 Proof of Proposition 3.2

From Eq. (9), we know $\tilde{X}_{\theta,n}^{\text{DRP}} \xrightarrow{a.s.} \tilde{X}_{\theta}$ as $n \rightarrow \infty$. Combining this with Eq. (25), we obtain $\text{SWD}(P, P^B) = \text{SWD}(\hat{P}_n, \hat{P}_n^B) = \text{SWD}(\hat{P}_{n,\text{DRP}}, \hat{P}_{n,\text{DRP}}^B)$ as $n \rightarrow \infty$.

G Effect of α_n in $\tilde{X}_{\theta}^{\text{DRP}}$

As discussed in Section 3.2, obtaining the swap property while maintaining low reconstructability at the sample level is empirically challenging. To address this, dependency regularization perturbation (DRP) is introduced. In this section, we evaluate the effect of α_n in $\tilde{X}_{\theta}^{\text{DRP}}$ in Eq. (9) on feature selection performance. Results are summarized in Figure 9 for five synthetic datasets with the sample size $n = 2000$.

When α_n is decreased, we observe an increase in power. However, the FDR exhibits a bowl-shaped pattern, consistent with the findings of [54]: introducing the permuted X_{rp} reduces reconstructability, thereby increasing power. However, an overly dominant X_{rp} compromises the swap property, resulting in higher FDRs. Based on our hyperparameter search and the results in Figure 9, we recommend choosing α_n within the range of 0.4 to 0.5.

H Implementation Details

H.1 Training

To fit models, we first split datasets of X into training and validation sets with an 8:2 ratio. The training sets are used for model optimization, and the validation sets are used for early stopping based on the validation loss, with a patience period of 6. Since knockoffs are not unique [11], testing sets are not required. To evaluate DeepDRK's performance, we compare it with four state-of-the-art (SOTA) deep-learning-based knockoff generation models, focusing on non-parametric data. Specifically, we

Parameter	Value	Parameter	Value
S_ω Learning Rate	1×10^{-3}	$\tilde{X}_\theta$ Learning Rate	1×10^{-5}
Dropout Rate	0.1	# of Epochs	200
Batch Size	64	λ_1	30.0
λ_2	1.0	λ_3	20.0
Early Stop Tolerance	6	α_n	0.5

Table 3: Training configuration.

include Deep Knockoff [47]¹², DDLK [55]¹³, KnockoffGAN [27]¹⁴, and sRMMD [38]¹⁵. We use the recommended hyperparameter settings for each model, with the only difference being the number of training epochs, set to 200 for consistency across models.

We follow the model configuration in Table 3 to optimize DeepDRK. The architecture of the swappers S_ω is based on [55]. Both the swappers and $\tilde{X}_\theta$ are trained using the AdamW optimizer [36]. During training, we alternately optimize $\tilde{X}_\theta$ and the swappers S_ω , updating weights θ three times for each update of weights ω . This training scheme is similar to GAN training [21], though without discriminators. We apply early stopping to prevent overfitting. A pseudocode of the optimization is provided in Appendix B.2. In experiments, we set $\alpha_n = 0.5$ universally as the dependency regularization coefficient due to its consistent performance¹⁶. A discussion on the effect of α_n is provided in Appendix G.

Once trained, models generate the knockoff $\tilde{X}$ given X data. The generated $\tilde{X}$ is then combined with X for feature selection. Experiments are conducted on a single NVIDIA V100 16GB GPU.

H.2 Feature Selection

Once $\tilde{X}$ is obtained, feature selection is performed in three steps. We first concatenate X and $\tilde{X}$ to form an $n \times 2p$ design matrix $(X, \tilde{X})$. In the second step, we use Ridge regression to get the estimated regression coefficients $\{\hat{\beta}_j\}_{j=1}^{2p}$ from Y and $(X, \tilde{X})$. Finally, we compute the knockoff statistics $w_j = |\hat{\beta}_j| - |\hat{\beta}_{j+p}|$, for $j = 1, 2, \dots, p$, and then select features using the threshold defined in Eq. (3). We set $q = 0.1$ as the FDR threshold, following its common use in other knockoff-based feature selection studies [47, 38]. Each experiment is repeated 600 times, with results reported as the average power and FDR over these 600 repetitions.

H.3 Data Preparation

H.3.1 Weights for the Gaussian mixture models

Table 4 includes 10 sets of (π_1, π_2, π_3) for the Gaussian mixture models.

H.3.2 Preparation of the RNA Data

We first normalize the raw data X to value range $[0, 1]$ and then standardize it to have zero mean and unit variance. Y is synthesized according to X . We consider two different ways of synthesizing Y . The first, denoted as “Linear”, is similar to the previous setup in the full synthetic case with $Y \sim \mathcal{N}(X^T \beta, 1)$ and $\beta \sim \frac{p}{12.5 \cdot \sqrt{N}} \cdot \text{Rademacher}(0.5)$. For the second, denoted as “Tanh”, the response Y is

Set No.	π_1	π_2	π_3
1	0.562	0.384	0.054
2	0.430	0.168	0.402
3	0.317	0.324	0.359
4	0.316	0.388	0.296
5	0.439	0.488	0.073
6	0.314	0.041	0.645
7	0.656	0.282	0.062
8	0.200	0.300	0.500
9	0.500	0.300	0.200
10	0.333	0.333	0.333

Table 4: 10 sets of (π_1, π_2, π_3) for the Gaussian mixture models.

¹²<https://github.com/mnesia/deepknockoffs>

¹³<https://github.com/rajesh-lab/ddlk>

¹⁴<https://github.com/vanderschaarlab/mlforhealthlabpub/tree/main/alg/knockoffgan>

¹⁵<https://github.com/ShoaibBinMasud/soft-rank-energy-and-applications>

¹⁶We present this value for its consistent performance; however, it may not be the globally optimal value. Future work may improve on α_n 's design.

Reference Type	Metabolite	Source	Meatbolite	Source
PubChem	palmitate	CID: 985	taurocholate	CID: 6675
	cholate	CID: 221493	p-hydroxyphenylacetate	CID: 127
	linoleate	CID: 5280450	deoxycholate	CID: 222528
	taurochenodeoxycholate	CID: 387316		
Publications	12.13-diHOME	[7]	dodecanedioate	[7]
	arachidonate	[7]	eicosatrienoate	[7, 5]
	eicosadienoate	[7]	docosapentaenoate	[7, 5]
	tauroolithocholate	[7]	salicylate	[7]
	saccharin	[7]	1.2.3.4-tetrahydro-beta-carboline-1.3-dicarboxylate	[7]
	oleate	[5]	arachidate	[5]
	glycocholate	[5]	chenodeoxycholate	[5]
	phenyllactate	[38, 31]	glycolithocholate	[5]
	urobilin	[38, 46]	caproate	[38, 32]
	hydrocinnamate	[38, 29]	myristate	[38, 18]
	adrenate	[38, 35]	olmesartan	[38, 48]
	tetradecanedioate	[56, 39]	hexadecanedioate	[56, 39]
	oxypurinol	[8]	porphobilinogen	[40]
	caprate	[52, 53]	undecanedionate	[32, 58]
	stearate	[2, 5]	oleanate	[44]
	glycochenodeoxycholate	[50]	sebacate	[32]
	nervonic acid	[57]	lithocholate	[5]
Preprints	alpha-muricholate	[42]	tauro-alpha-muricholate/tauro-beta-muricholate	[42]
	17-methylstearate	[42]	myristoleate	[42]
	taurodeoxycholate	[42]	ketodeoxycholate	[42]

Table 5: IBD-associated metabolites that are supported by the literature. This table includes all 47 referenced metabolites for the IBD case study. Each metabolite is supported by one of the three sources: PubChem, peer-reviewed publications or preprints. For PubChem case, we report the PubChem reference ID (CID), and for the other two cases we report the publication references.

generated following the expression:

$$\begin{aligned}
k &\in [m/4] \\
\varphi_k^{(1)}, \varphi_k^{(2)} &\sim \mathcal{N}(1, 1) \\
\varphi_k^{(3)}, \varphi_k^{(4)}, \varphi_k^{(5)} &\sim \mathcal{N}(2, 1) \\
Y \mid X &= \epsilon + \sum_{k=1}^{m/4} \varphi_k^{(1)} X_{4k-3} + \varphi_k^{(3)} X_{4k-2} \\
&\quad + \varphi_k^{(4)} \tanh\left(\varphi_k^{(2)} X_{4k-1} + \varphi_k^{(5)} X_{4k}\right),
\end{aligned} \tag{26}$$

where ϵ follows the standard normal distribution and the 20 covariates are sampled uniformly.

H.3.3 Metabolite Information for the IBD Study

Here we provide the implementation details for the experiments described in Section 4.3. In Table 5, we provide all 47 referenced metabolites based on our comprehensive literature review.

I Additional Results

In this section, we include all results that are deferred from the main paper.

I.1 Swap Property

We use different metrics to empirically evaluate the swap property on the generated knockoff $\tilde{X}$ and the original data X according to Eq. (1). In this paper, three metrics are considered: mean discrepancy distance with linear kernel, or “MMD(Linear)” for

Abbreviation	Full Name
J+G	Copula: Joe & Gamma
C+G	Copula: Clayton & Gamma
C+E	Copula: Clayton & Exponential
J+E	Copula: Joe & Exponential
MG	Mixture of Gaussians

Table 6: The abbreviations of the names for the datasets.

	Dataset	MMD(Linear)	SWD ₁	SWD ₂
DDLK	J+G	2.68	0.18	0.08
	C+G	2.49	0.18	0.07
	C+E	1.80	0.15	0.05
	J+E	2.29	0.15	0.06
	MG	306.49	2.15	9.99
KnockoffGAN	J+G	7.01	0.15	0.08
	C+G	5.11	0.16	0.04
	C+E	0.52	0.06	0.01
	J+E	1.24	0.08	0.02
	MG	1.08	3.28	26.09
Deep Knockoff	J+G	13.09	0.21	0.11
	C+G	19.04	0.27	0.14
	C+E	6.65	0.18	0.07
	J+E	6.78	0.19	0.13
	MG	2770.00	8.98	196.54
DeepDRK (Ours)	J+G	0.47	0.13	0.06
	C+G	0.71	0.14	0.05
	C+E	0.23	0.09	0.02
	J+E	0.14	0.10	0.04
	MG	380	6.13	84.37
sRMMD	J+G	130.02	0.68	0.71
	C+G	175.74	0.73	0.96
	C+E	49.09	0.41	0.35
	J+E	33.24	0.35	0.29
	MG	3040	8.51	142.99

Table 7: Evaluation of the swap property. This table empirically measures the swap property by three different metrics: MMD(Linear), SWD₁, and SWD₂. The evaluation considers all baseline models and all datasets in the synthetic dataset setup. For space consideration, we use abbreviations to indicate the name of the datasets. The full name can be found in Table 6.

short; sliced Wasserstein-1 distance (SWD₁); and sliced Wasserstein-2 distance (SWD₂). We measure the sample level distances between the vector that concatenates X and $\tilde{X}$ (i.e., $(X, \tilde{X})$) and the vector after randomly swapping the entries (i.e., $(X, \tilde{X})_{\text{swap}(B)}$). To avoid repetition, please refer to section 2.1 and Eq. (1) for the definition of notation. Empirically, it is time consuming to evaluate all subsets B of the index set $[p]$. As a result, we alternatively define a swap ratio $r_s \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The swap ratio controls the amount of uniformly sampled indices (i.e., the cardinality $|B| = r_s \cdot p$) in a subset of $[p]$. For any X and $\tilde{X}$ from the same experiment, 5 different subsets B are formed according to 5 different swap ratios. We report the average values over the swap ratios to represent the empirical quantification of the swap property. Results can be found in Table 7.

Clearly, compared to other models, the proposed DeepDRK achieves the smallest values in almost every entry across the three metrics and the first 4 datasets (i.e., J+G, C+G, C+E and J+E in Table 7). This explains why DeepDRK has lower FDRs as DeepDRK maintains the swap property relatively better than the baseline models (see results in Figure 2). Similarly, we observe that KnockoffGAN also achieves relatively small values, which leads to well-controlled FDRs compared to other baseline models. Overall, this verifies the argument in Candès et al. [11] that the swap property is important in guaranteeing FDR during feature selection.

However, we observe a difference for Gaussian mixture data. The proposed DeepDRK achieves the best performance in FDR control and power (see results in Figure 2), yet its swap property measured under the metrics in Table 7 is not the lowest. Despite this counter-intuitive observation, we want to highlight that it does not conflict with the argument in Candès et al. [11]. Rather, it supports our statement that the low reconstructability and the swap property cannot be achieved at the sample level (e.g., the free lunch dilemma in practice). After all, the swap property is not the only factor that determines FDR and power during feature selection.

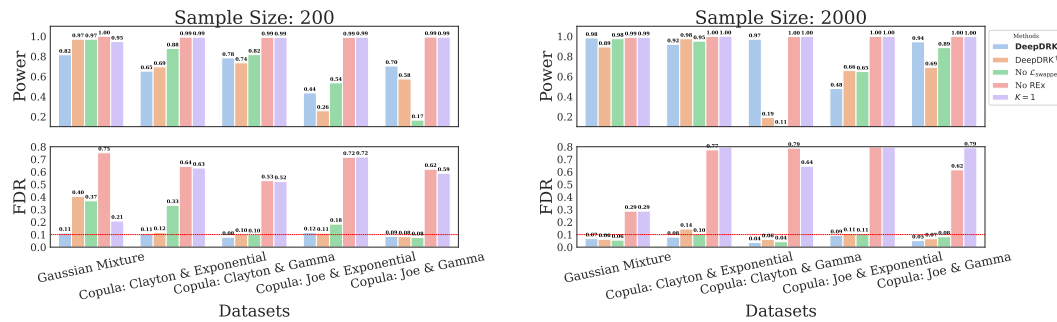


Figure 10: Power and FDR in the ablation studies. The red horizontal line indicates the 0.1 FDR threshold. DeepDRK[†] is the model with dependency regularized perturbation removed. K indicates the number of swappers.

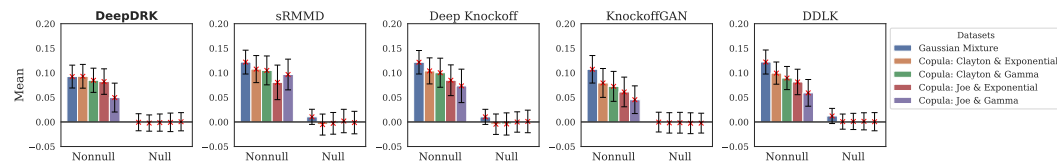


Figure 11: Knockoff statistics (w_j) for different knockoff models on the synthetic datasets. Each bar in the plot represents the mean of the null/nonnull knockoff statistics averaging on 600 experiments. The error bar indicates the standard deviation. The sample size is 2000.

I.2 Ablation Studies

In this subsection, we perform ablation studies on different terms introduced in Section 3.1.1, to show the necessity of designing these terms during the optimization for knockoffs. We consider the fully synthetic setup described in Section 4.1, and consider $n = 200$ and $n = 2000$. The distribution of β is $\frac{p}{15 \cdot \sqrt{N}} \cdot \text{Rademacher}(0.5)$. We test the following terms: 1. REx; 2. the number of swappers K ; 3. $\mathcal{L}_{\text{swapper}}$; 4. the dependency regularized perturbation (denoted as DeepDRK[†]). Five synthetic datasets are considered as before and we report the values for FDR and Power under each setup. The results are presented in Figure 10.

Figure 10 illustrates the clear drawbacks of using only a single swapper ($K = 1$) and the case without the loss term REx. In those two cases, we fail to control the FDR on all datasets. The REx term is essential, as it ensures that the adversarial attacks generated by different swappers are addressed simultaneously. $\mathcal{L}_{\text{swapper}}$ is also an important term, as it promotes a diverse adversarial environments. Without this term we observe an increase in FDR compared to DeepDRK or DeepDRK[†]. The difference between DeepDRK and DeepDRK[†] is more subtle as DeepDRK[†] has already obtained high quality knockoffs. However, we empirically observe that adding the perturbation further increases the power for some datasets without sacrificing the FDR controllability. To interpret these observations, we follow a similar procedure in Section 4.1 to study the distribution of the knockoff statistics. Results are included in Section I.3.

Overall, we verify that all terms are necessary components to achieve higher powers and controlled FDRs through the ablation studies.

I.3 Analysis on the distribution of knockoff statistics

We present the additional results on the means and standard deviations of both null and nonnull knockoff statistics under various experimental setups.

Figure 11 shows the results on w_j for $n = 2000$ and it complements the results on Power and FDR in Figure 2. We notice that all models have concentrated null features around zero and relatively high values for the nonnull knockoff statistics for the case with 2000 samples. This corresponds to consistently accurate FS results across all models and datasets, as shown in Figure 2.

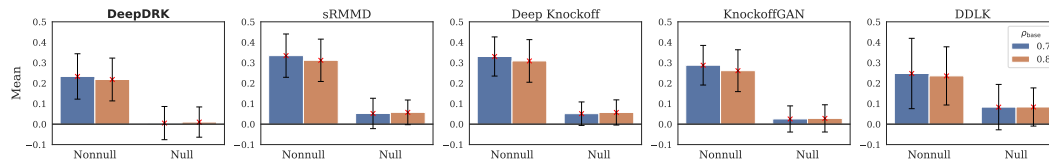


Figure 12: Knockoff statistics (w_j) for different models with the increased correlation of the Gaussian mixture data. Each bar in the plot represents the mean of the null/nonnull knockoff statistics averaging on 600 experiments. The error bar indicates the standard deviation. The sample size is 200.

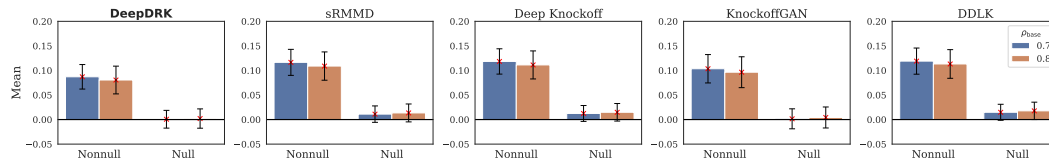


Figure 13: Knockoff statistics (w_j) for different models with the increased correlation of the mixture of Gaussians data. Each bar in the plot represents the mean of the null/nonnull knockoff statistics averaging on 600 experiments. The error bar indicates the standard deviation. The sample size considered is 2000.

Figure 12 and Figure 13 show the results for the Gaussian mixture data with increased correlations ($\rho_{\text{base}} = 0.7$ and 0.8) for $n = 200$ and $n = 2000$ respectively. The figures complement the FDR-power results in Figure 3. It is clear for both $n = 200$ and $n = 2000$ cases, all models experience an increase in FDR and a decrease in power. This phenomenon can be reflected by the positive shifts of the knockoff statistics for the null features in Figure 12 and Figure 13. However, DeepDRK significantly controls the shifts, achieving the best results with the lowest FDRs and comparable power as shown in Figure 3.

In addition, we also compare the knockoff statistics for the models considered in the ablation studies in I.2. The results for both $n = 200$ and $n = 2000$ are in Figure 14 and Figure 15. Although the null knockoff statistics for DeepDRK, DeepDRK[†] and “No $\mathcal{L}_{\text{swapper}}$ ” models concentrate symmetrically around zero, the heights of the nonnull knockoff statistics for DeepDRK are the highest, resulting higher power. And because some nonnull knockoff statistics have small values for the DeepDRK[†] and “No $\mathcal{L}_{\text{swapper}}$ ” cases, we also expect them to have higher FDRs (see Figure 10). On the other hand, compared to DeepDRK, the models of “No REX” and “ $K = 1$ ” experience clear positive shifts for the null knockoff statistics, leading to higher FDRs during FS (see Figure 10).

I.4 Training time

We consider evaluating and comparing model training runtime in Table 8 with the (2000, 100) setup, as it is common in the existing literature. Although DeepDRK is not the fastest among the compared models, the time cost—7.35 minutes—is still short, especially when the performance is taken into account.

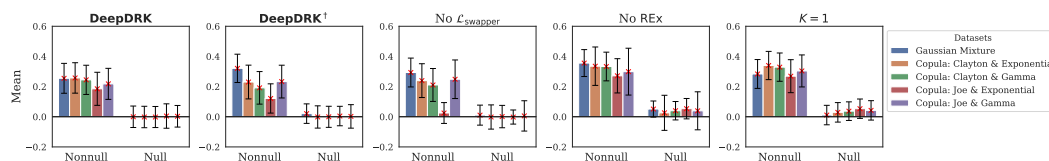


Figure 14: Knockoff statistics (w_j) for different models in ablation studies. Each bar in the plot represents the mean of the null/nonnull knockoff statistics averaging on 600 experiments. The error bar indicates the standard deviation. K refers to the number of swappers. The sample size is 200.

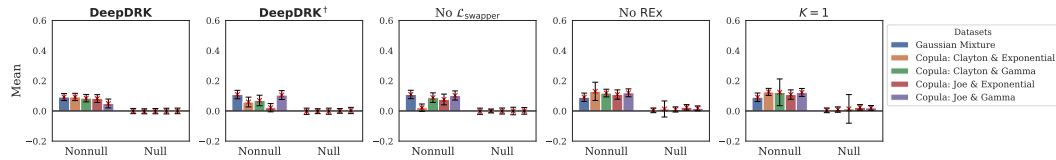


Figure 15: Knockoff statistics (w_j) for different models in ablation studies. Each bar in the plot represents the mean of the null/nonnull knockoff statistics averaging on 600 experiments. K refers to the number of swappers. The sample size considered is 2000.

DeepDRK	Deep Knockoff	sRMMD	KnockoffGAN	DDLK
7.35 min	1.08 min	6.38 min	10.52 min	53.63 min

Table 8: The training time for the models with $n = 2000$ and $p = 100$. The batch size is 64 and the models are all trained with 100 epochs.

I.5 Additional Results for the IBD Study

Here we provide the supplementary information for the experimental results described in Section 4.3. In Table 9, we provide the list of identified metabolites by each of the considered models. This table provides additional information for Table 2 in the main paper which only includes metabolite counts due to limited space.

Metabolite	DeepDRK	Deep Knockoff	sRMMD	KnockoffGAN	DDLK
12.13-diHOME				*	
9.10-diHOME					
caproate	*	*	*		*
hydrocinnamate					
mandelate					
3-hydroxyoctanoate					
caprate					
indoleacetate					*
3-hydroxydecanoate					
dodecanoate				*	
undecanedionate	*			*	
myristoleate					
myristate					
dodecanedioate				*	
pentadecanoate					
hydroxymyristate					
palmitoleate					
palmitate				*	
tetradecanedioate		*			
10-heptadecenoate					
2-hydroxyhexadecanoate					
alpha-linolenate					*
linoleate					
oleate					
stearate					*
hexadecanedioate		*		*	*
10-nonadecenoate					
nonadecanoate					
17-methylstearate	*	*			*
eicosapentaenoate	*	*			*
arachidonate	*	*		*	*
eicosatrienoate	*	*			*
eicosadienoate	*	*	*		*
eicosenoate					
arachidate				*	
phytanate					
docosahexaenoate	*	*			*
docosapentaenoate	*	*		*	*
adrenate	*	*	*	*	*
13-docosenoate					
eicosanedioate	*	*			
oleanate					
masilinate					
lithocholate	*				
chenodeoxycholate					
deoxycholate	*			*	*
hyodeoxycholate/ursodeoxycholate					
ketodeoxycholate	*				
alpha-muricholate	*				
cholate		*			
glycolithocholate	*				
glycochenodeoxycholate					
glycodeoxycholate					
glycoursodeoxycholate					
glycocholate					
tauroolithocholate					*
taurochenodeoxycholate					
taurodeoxycholate					
taurohyodeoxycholate/tauroursodeoxycholate					
tauro-alpha-muricholate/tauro-beta-muricholate		*			*
taurocholate					
salicylate	*	*	*	*	
saccharin				*	
azelate					*
sebacate	*				*
carboxybuprofen					
olmesartan					
1.2.3.4-tetrahydro-beta-carboline-1.3-dicarboxylate					
4-hydroxystyrene		*		*	*
acetytyrosine					
alpha-CEHC					
carnosol					*
oxypurinol					
palmitoylethanolamide					
phenyllactate	*	*	*		*
p-hydroxyphenylacetate	*	*			*
porphobilinogen	*				
urobilin	*	*		*	*
nervonic acid					
oxymetazoline	*	*			*

Table 9: A list of literature-supported metabolites out of a total of 80 candidates. “*” indicates the important metabolites marked by the corresponding algorithms.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist",**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: We clearly describe the scope and the objective of the paper in the abstract and the introduction sections (also in the related work section).

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The limitation is described in the footnote 15 on page 20.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Please refer to the method section and the appendix for the complete proof. We provide assumptions in the method section.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We run the experiments multiple times for reproducibility check.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: We provide open code access.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: They are included in the experiment section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: Results shown in the experiment section are repeated for 600 times.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: All experiments are performed with a single NVIDIA V100 GPU. It is described in the experiment section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: N/A

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper focuses purely on methodology. How to use this method, and to serve what purpose is completely dependent on people who use it (see the conclusion section).

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: It is an unrelated question to this paper.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: The code is open-sourced on GitHub, with necessary licenses.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: Code is the only asset. Since it is associated with the method (see the method section), the answer is yes.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: It is unrelated.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: It is unrelated.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.