Deep Knockoffs

Yaniv Romano*[†], Matteo Sesia*[†], Emmanuel J. Candès[†][‡]

November 15, 2018

Abstract

This paper introduces a machine for sampling approximate model-X knockoffs for arbitrary and unspecified data distributions using deep generative models. The main idea is to iteratively refine a knockoff sampling mechanism until a criterion measuring the validity of the produced knockoffs is optimized; this criterion is inspired by the popular maximum mean discrepancy in machine learning and can be thought of as measuring the distance to pairwise exchangeability between original and knockoff features. By building upon the existing model-X framework, we thus obtain a flexible and *model-free* statistical tool to perform controlled variable selection. Extensive numerical experiments and quantitative tests confirm the generality, effectiveness, and power of our deep knockoff machines. Finally, we apply this new method to a real study of mutations linked to changes in drug resistance in the human immunodeficiency virus.

1 Introduction

1.1 Motivation

Model-X knockoffs [1] is a new statistical tool that allows the scientist to investigate the relationship between a response of interest and hundreds or thousands of explanatory variables. In particular, model-X knockoffs can be used to identify a subset of important variables from a larger pool that could potentially explain a phenomenon under study while rigorously controlling the false discovery rate [2] in very complex statistical models. While this methodology does not require any knowledge of how the response depends on the values of the features, the correctness of the inferences rests entirely on a precise description of the distribution of the explanatory variables, which are assumed to be random. This makes model-X knockoffs well adapted to situations in which good models are available to describe the joint distribution of the features, as in genome-wide association studies [3] where hidden Markov models are widely used to describe patterns of genetic variation. To apply the knockoffs approach in a broad set of applications, however, we would need flexible tools to construct knockoff variables from sampled data in settings where we do not have reliable prior knowledge about the distribution of the covariates but perhaps sufficiently many labeled or unlabeled samples to 'learn' this distribution to a suitable level of approximation. These conditions are realistic because the construction of model-X knockoffs only depends on the explanatory variables whose unsupervised observations may be abundant. For example, even though the genome-wide association analysis of a rare

*These authors are listed in alphabetical order.

[†] Department of Statistics, Stanford University, Stanford, CA 94305, U.S.A.

[‡] Department of Mathematics, Stanford University, Stanford, CA 94305, U.S.A.

disease may contain a relatively small number of subjects, the genetic variants for other individuals belonging to the same population can be gathered from different studies.

The goal of this paper is simply stated: to extend the applicability of the knockoffs framework as to make it practically model-free and, therefore, widely applicable. This is achieved by taking advantage of important recent progress in machine learning, which is repurposed to harness the information contained in large unsupervised datasets to sample approximate model-X knockoffs. The ultimate outcome is a set of sensible and flexible tools for model-free controlled variable selection that can help alleviate the crucial irreproducibility issues afflicting many areas of science and data analysis [4–7]. A preview of our contribution is sketched below, while the technical details are postponed to later sections.

1.2 Our contribution

Given independent copies of $X = (X_1, \ldots, X_p) \in \mathbb{R}^p$ from some unknown distribution P_X, we seek to construct a random generator of valid knockoffs $\tilde{X} = (\tilde{X}_1, \ldots, \tilde{X}_p)$ such that the joint law of $(X, \tilde{X})$ is invariant under the swapping of any X_j and $\tilde{X}_j$ for each $j \in \{1, \ldots, p\}$ (see Section 2 for details). Concretely, the machine takes the data X as input and generates $\tilde{X}$ through a mapping $f_\theta(X, V)$, where V is random noise, and f_θ is a deep neural network. The parameters of the network are fitted on multiple observations of X to optimize a loss function that quantifies the extent to which $\tilde{X}$ is a good knockoff copy of X. This goal is related to the classical problem of learning generative models; however, the challenge here is unusual since only X is accessible while no sample from the target distribution $P_{\tilde{X}|X}$ is available. Fortunately, the existing methods of deep generative modeling reviewed in Section 3 can be suitably repurposed, as we shall see in Section 4. Furthermore, the lack of uniqueness of the target distribution raises an additional question. Intuitively, this ambiguity should be resolved by making $\tilde{X}$ as different as possible from X, since a trivial copy—setting $\tilde{X} = X$—would satisfy the required symmetry without being of any practical use for variable selection. Our approach generalizes the solution described in [1], which relies on the simplifying assumption that X can be well-described as a multivariate Gaussian vector. In the context of deep generative models, the analogous idea consists of training a machine that optimizes the compatibility of the first two moments of $(X, \tilde{X})$ while keeping the strength of the pairwise correlations between X_j and $\tilde{X}_j$ for each $j \in \{1, \ldots, p\}$ under control. By including in the loss function an additional term that promotes the matching of higher moments, we will show that one can move beyond the second-order approximation towards a model-free knockoff generator. The effectiveness of deep knockoff machines can be quantitatively measured using the goodness-of-fit diagnostics presented in Section 5, as shown empirically by the results of our numerical experiments (Section 6) and data analysis (Section 7). The algorithms described in this paper have been implemented in Python and the corresponding software is available from https://web.stanford.edu/group/candes/deep-knockoffs/.

1.3 Related work

The main idea of using knockoffs as negative control variables was originally devised in the context of linear regression setting with a fixed design matrix [8]. The generation of model-X knockoffs beyond the settings considered in [1] and [3] has also been tackled in [9], which extends the results for hidden Markov models to a broader class of Bayesian networks. More recent advances in the framework of knockoffs include the work of [10–12], while some interesting applications can be found in [13–15]. Very recently, deep generative models have independently been suggested as a procedure for sampling knockoffs in [16]; there, the approach focuses on adversarial rather than moment matching networks. Even though the fundamental aims coincide and the solutions are related, our machine differs profoundly by design and it offers a more direct connection with existing work on second-order knockoffs. Also, it is well known that generative adversarial networks are difficult to train [17], while moment matching is a simpler task [18, 19]. Since the approach of [16] requires simultaneously training four different and interacting neural networks, we expect that a good configuration for our machine should be faster to learn and require less tuning. This may be a significant advantage since the ultimate goal is to make knockoffs easily accessible to researchers from different fields. A computationally lighter alternative is proposed in [20], which relies on the variational autoencoder [21] to generate knockoff copies. Since our work was developed in parallel[1] to that of [16, 20], we are not including these recent proposals in our simulation studies. Instead, we will compare our method to well-established alternatives.

2 Model-X knockoffs

Since the scope of our work depends on properties of model-X knockoffs, we begin by rehearsing some of the key features of the existing theory. For any $X \in \mathbb{R}^p$ sampled from a distribution P_X, the random vector $\tilde{X} \in \mathbb{R}^p$ is said to be a knockoff copy of X [1] if the joint law of $(X, \tilde{X})$ obeys

$$(X, \tilde{X}) \overset{d}{=} (X, \tilde{X})_{\text{swap}(j)} \quad \text{for each } j \in \{1, \dots, p\}; \tag{1}$$

here, the symbol $\overset{d}{=}$ indicates equality in distribution and $(\cdot)_{\text{swap}(j)}$ is defined as the operator swapping X_j with $\tilde{X}_j$; if $p = 3$ and $j = 2$, $(X_1, X_2, X_3, \tilde{X}_1, \tilde{X}_2, \tilde{X}_3)_{\text{swap}(j)} = (X_1, \tilde{X}_2, X_3, \tilde{X}_1, X_2, \tilde{X}_3)$. Knockoffs play a key role in controlled variable selection, by serving as negative controls that allow one to estimate and limit the number of false positives in the variable selection problem defined below.

Consider n observations $\{X^i, Y^i\}_{i=1}^n$, with each $X^i = (X_1^i, \dots, X_p^i) \in \mathbb{R}^p$ assumed to be drawn independently from a known P_X, and the associated label $Y^i \in \mathbb{R}$ drawn from an unknown conditional distribution $P_{Y|X}$. The goal is to identify a subset of important components of X that affect Y. In order to state this objective more precisely, one refers to X_j as unimportant if

$$Y \perp\!\!\!\perp X_j \mid X_{-j},$$

where X_{-j} indicates the remaining $p-1$ variables after X_j is excluded. The true null hypotheses $\mathcal{H}_0$ is the set of all variables that are unimportant; in words, X_j is not important if it is conditionally independent of the response Y once we know the value of X_{-j}. Put differently, X_j is not important if it does not provide any additional information about Y beyond what is already known. While searching for a subset $\hat{\mathcal{S}}$ that includes the largest possible number of important variables in $\mathcal{H}_1 = \{1, \dots, p\} \setminus \mathcal{H}_0$, one wishes to ensure that the false discovery rate,

$$\text{FDR} = \mathbb{E}\left[\frac{|\hat{\mathcal{S}} \cap \mathcal{H}_0|}{|\hat{\mathcal{S}} \vee 1|}\right],$$

remains below a nominal level $q \in (0,1)$, e.g. $q = 0.1$. The false discovery rate is thus defined as the expected fraction of selected variables that are false positives.

The approach of [1] provably controls the false discovery rate without placing any restrictions on the conditional likelihood of $Y \mid X$, which can be arbitrary and completely unspecified. The first step in their method consists of generating a knockoff copy $\tilde{X}$ for each available sample of X, before looking at Y, such that both (1) is satisfied and $Y \perp\!\!\!\perp \tilde{X} \mid X$. Some measures of feature importance Z_j and $\tilde{Z}_j$ are then evaluated

[1] The results of this paper were first discussed at the University of California, Los Angeles, during the Green Family Lectures on September 27, 2018.

for each X_j and $\tilde{X}_j$, respectively. For this purpose, almost any available method from statistics and machine learning can be applied to the vector of labels $\mathbf{Y}$ and the augmented data matrix $[\mathbf{X}, \tilde{\mathbf{X}}] \in \mathbb{R}^{n \times 2p}$, with the only fundamental rule that the original variables and the knockoffs should be treated equally; this is saying that the method should not use any information revealing which variable is a knockoff and which is not. Examples include sparse generalized linear models [1, 3], random forests [15], support vector machines and deep neural networks [9, 10]. Each pair of Z_j and $\tilde{Z}_j$ is then combined through an antisymmetric function into the statistics W_j, e.g. $W_j = Z_j - \tilde{Z}_j$. By construction, a large and positive value of W_j suggests evidence against the jth null hypothesis, while unimportant variables are equally likely to be positive or negative. Under this choice of W_j, it can be shown that exact control of the false discovery rate below the nominal level q can be obtained by selecting $\hat{\mathcal{S}} = \{j : W_j \geq \tau_q\}$, where

$$\tau_q = \min \left\{ t > 0 : \frac{1 + |\{j : W_j \leq -t\}|}{|\{j : W_j \geq t\}|} \leq q \right\}.$$

The numerator in the expression above can be understood as a conservative estimate of the number of false positives above the fixed level t. This adaptive significance threshold is that first proposed in the knockoff filter of [8], while the choice of the test statistics W_j may be different [1].

The validity of the false discovery rate control relies entirely on the exact knowledge of P_X and our ability to generate $\tilde{X}$ satisfying (1). Even though procedures that can sample exact knockoff copies have been previously derived for a few special classes of P_X such as multivariate Gaussian distributions [1] and hidden Markov models [3], the general case remains algorithmically challenging. This difficulty arises because (1) is much more stringent than a first look may suggest. For instance, obtaining new independent samples from P_X or permuting the rows of the data matrix would only ensure that (X_1, X_2) is equal in distribution to $(\tilde{X}_1, \tilde{X}_2)$, while the analogous result would not hold between (X_1, X_2) and $(X_1, \tilde{X}_2)$. At the same time, the latter property is crucial since a null variable and its knockoff must be able to explain on average the same fraction of the variance in the response. The practical approximate solution described in [1] consists of relaxing the condition in (1) as to match only the first two moments of the distributions on either side. In this weaker sense, $\tilde{X}$ is thus said to be a *second-order knockoff* copy of X if the two random vectors have the same expected value and their joint covariance matrix is equal to

$$\operatorname{Cov}\left[(X, \tilde{X})\right] = \begin{bmatrix} \Sigma & \Sigma - \operatorname{diag}(s) \\ \Sigma - \operatorname{diag}(s) & \Sigma \end{bmatrix}, \tag{2}$$

where Σ is the covariance matrix of X under P_X and s is any p-dimensional vector selected in such a way that the matrix in the right-hand side is positive semidefinite. The role of s is to make $\tilde{X}$ as uncorrelated with X as possible, in order to increase statistical power during variable selection. Therefore, the value of s is typically chosen to be as large as possible [1]. The weaker form of exchangeability in (2) is reminiscent of the notion of fixed-design knockoffs from [8] and it can be practically implemented by approximating the distribution of X as multivariate Gaussian [1]. This approximation often works well in practice, even though it is in principle insufficient to guarantee control of the false discovery rate under the general conditions of the model-X framework [22]. In this paper, we build upon the work of [1] and [22] to obtain higher-order knockoffs that can achieve a better approximation of (1) using modern techniques from the field of deep generative models.

3 Deep generative models

Replicating the underlying distribution of a data source is an essential task of statistical machine learning that can be broadly described as follows. Given n independent p-dimensional samples $\{X^i\}_{i=1}^n$ from an unknown distribution P_X, a generative model approximating the true P_X is sought in order to synthesize new observations that could plausibly belong to the training set, while being sufficiently different to be non-trivial. Several well known techniques have been developed to tackle this problem, some of which are based on hidden Markov models [23], Gaussian mixture models [24] or Boltzmann machines [25]. In recent years, such traditional approaches have been largely replaced by neural networks, with two popular examples being variational autoencoders [21, 26–28] and generative adversarial networks [29–35]. These are based on a parametric non-linear function $f_\theta(V)$ that maps an input noise vector V to the sample domain of X. The parameters in θ represent the collection of weights and biases defining the neural network and they need to be learned from the available data. The function thus defined is deterministic for any fixed realization of the noise and, with an appropriate choice of θ, it propagates and transforms the noise in V to obtain a random variable $f_\theta(V)$ approximately distributed as X.

Training deep generative models is computationally difficult, and considerable effort has been dedicated to the development of practical algorithms that can find good solutions. For instance, the popular variational method, which lies at the heart of the autoencoder in [21], proceeds by maximizing a traceable lower bound on the log-likelihood of the training data. In contrast, generative adversarial networks strive to minimize the inconsistencies of the generated samples with the original ones, by formulating the learning task as a two-player game [29]. As the generator $f_\theta(V)$ attempts to produce realistic samples, an antagonistic discriminator tries to recognize them. Since the discriminator is defined as a deep binary classifier with a differentiable loss function, the two networks can be simultaneously trained by gradient descent, until no further gain can be made on either side. Even though generative adversarial networks have enjoyed a great deal of success [29–35], non-convex minimax optimization is notoriously complex [17]. More recent alternatives mitigate the issue by replacing the classifier with a simpler measure of the distance between the distributions of the original and the simulated samples [18, 19, 36–39]. The remaining part of this section is dedicated to reviewing the basics of some of the latter approaches, upon which we will begin to develop a knockoff machine.

The discriminator component of a deep generative model faces the following challenge. Given two sets of independent observations $\{X^i\}_{i=1}^n$ and $\{Z^i\}_{i=1}^n$, respectively drawn from some unknown distributions P_X and P_Z, it must be verified whether $P_X = P_Z$. This multivariate two-sample problem has a long history in the statistics literature and many non-parametric tests have been proposed to address it [40–46]. In particular, the work of [45] introduced a test statistic, called the maximum mean discrepancy, whose desirable computational properties have inspired the development of generative moment matching networks [18, 19]. The relevant key idea is to quantify the discrepancy between the two distributions in terms of the largest difference in expectation between $\phi(X)$ and $\phi(Z)$, over functions ϕ mapping the random variables into the unit ball of a reproducing kernel Hilbert space [45]. Fortunately, this abstract characterization can be made explicit with the kernel trick [45], leading to the practical utilization described below.

Let X, X', Z, Z' be independent samples drawn from P_X and P_Z, respectively, and define the maximum mean discrepancy between P_X and P_Z as

$$\mathcal{D}_{\text{MMD}}(P_X, P_Z) = \mathbb{E}_{X,X'}\left[k(X, X')\right] - 2\mathbb{E}_{X,Z}\left[k(X, Z)\right] + \mathbb{E}_{Z,Z'}\left[k(Z, Z')\right], \tag{3}$$

where k is a kernel function. If the characteristic kernel of a reproducing kernel Hilbert space [45] is used, it can be shown that the quantity in (3) is equal to zero if and only if $P_X = P_Z$. Concretely, valid choices of k include the common Gaussian kernel, $k(X, X') = \exp\{-\|X - X'\|_2^2/(2\xi^2)\}$, with bandwidth parameter $\xi > 0$, and mixtures of such. Furthermore, the maximum mean discrepancy is always non-negative and it can be estimated from finite samples $\mathbf{X}, \mathbf{Z} \in \mathbb{R}^{n\times p}$ in an unbiased fashion via

$$\widehat{\mathcal{D}}_{\text{MMD}}(\mathbf{X}, \mathbf{Z}) = \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i} k(X^i, X^j) - \frac{2}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} k(X^i, Z^j) + \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i} k(Z^i, Z^j), \tag{4}$$

see [45]. Since the expression in (4) is easily computable and differentiable, it can serve as the objective function of a deep generative model, effectively replacing the discriminator required by generative adversarial networks [18, 19]. The generator is then trained on $\mathbf{X}$ to produce samples $\mathbf{Z}$ that minimize (4), by applying the standard techniques of gradient descent. This idea can also be repurposed to develop a knockoff machine, as discussed in the next section.

4 Deep knockoff machines

4.1 Overview

A knockoff machine is defined as a random mapping f_θ that takes as input a random $X \in \mathbb{R}^p$, an independent noise vector $V \sim \mathcal{N}(0, I) \in \mathbb{R}^p$ and returns an approximate knockoff copy $\tilde{X} = f_\theta(X, V) \in \mathbb{R}^p$. The machine is characterized by a set of parameters θ and it should be designed such that the joint distribution of $(X, \tilde{X})$ deviates from (1) as little as possible. If the original variables follow a multivariate Gaussian distribution, i.e. $X \sim \mathcal{N}(0, \Sigma)$, a family of machines generating exact knockoffs is given by

$$f_\theta(X, V) = X - X\Sigma^{-1}\text{diag}\{s\} + \left(2\text{diag}\{s\} - \text{diag}\{s\}\Sigma^{-1}\text{diag}\{s\}\right)^{1/2} V, \tag{5}$$

for any choice of the vector s that keeps the matrix multiplying V positive-definite [1]. In practice, the value of s is typically determined by solving a semi-definite program [1], see Section 4.5. By contrast, the algorithm for sampling knockoff copies of hidden Markov models in [3] cannot be easily represented as an explicit function f_θ. This difficulty should be expected for various other choices of P_X, and an analytic derivation of f_θ seems intractable in general.

In order to develop a flexible machine that can sample knockoffs for arbitrary and unknown distributions P_X, we assume f_θ to take the form of a deep neural network, as described in Section 4.5. The values of its parameters will be estimated on the available observations of X by solving a stochastic optimization problem. An overview of our approach is visually presented in Figure 1 and it can be summarized as follows: the machine is provided with n realizations of the random vector X, independently sampled from an unknown underlying distribution P_X. For any fixed configuration of θ, each $\tilde{X}^i$ is computed as a function of the corresponding input X^i and the noise V^i, for $i \in \{1, \dots, n\}$. The latter (V^i) is independently resampled for each observation and each time the machine is called. A scoring function J examines the empirical distribution of $(X, \tilde{X})$ and quantifies its compliance with the exchangeability property in (1). After each such iteration, the parameters θ are updated in the attempt to improve future scores. Ideally, upon successful completion of this process, the machine should be ready to generate approximate knockoff copies $\tilde{X}$ for new observations of X drawn from the same P_X. A specific scoring function that can generally lead to high-quality knockoffs will be defined below.

4.2 Second-order machines

We begin by describing the training of a special knockoff machine that is interesting for expository purposes. Suppose that instead of requiring the joint distribution of $(X, \tilde{X})$ to satisfy (1), we would be satisfied with obtaining second-order knockoffs. In order to incentivize the machine to produce $\tilde{X}$ such that $\mathbb{E}[X] = \mathbb{E}[\tilde{X}]$ and the joint covariance matrix of $(X, \tilde{X})$ satisfies (2), we consider a simple loss function that computes a differentiable measure of its compatibility with these requirements. For the sake of notation, we let $\hat{G}$

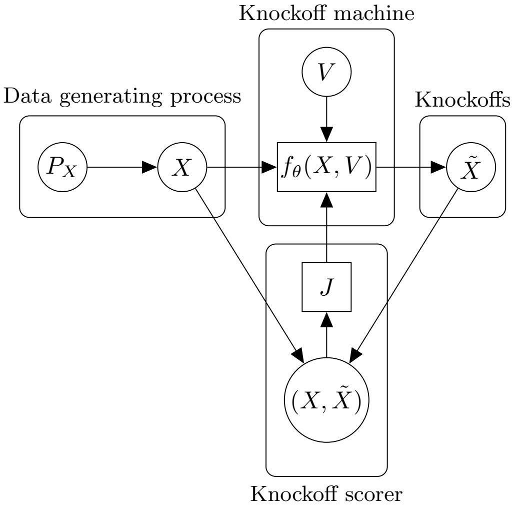


Figure 1: Schematic representation of the learning mechanism of a knockoff machine. The arrows indicate the flow of information between the source of data, the machine and the knockoff scoring function.

indicate the empirical covariance matrix of $(X, \tilde{X}) \in \mathbb{R}^{2p}$, which takes the following block form:

$$\hat{G} = \begin{bmatrix} \hat{G}_{XX} & \hat{G}_{X\tilde{X}} \\ \hat{G}_{X\tilde{X}} & \hat{G}_{\tilde{X}\tilde{X}} \end{bmatrix}. \tag{6}$$

Above, $\hat{G}_{XX}, \hat{G}_{\tilde{X}\tilde{X}} \in \mathbb{R}^{p \times p}$ are the empirical covariance matrices of $X, \tilde{X}$, respectively. Then we define

$$J_{\text{second-order}}(\mathbf{X}, \tilde{\mathbf{X}}) = \lambda_1 \frac{\|\frac{1}{n}\sum_{i=1}^n (X^i - \tilde{X}^i)\|_2^2}{p} + \lambda_2 \frac{\|\hat{G}_{XX} - \hat{G}_{\tilde{X}\tilde{X}}\|_F^2}{\|\hat{G}_{XX}\|_F^2} + \lambda_3 \frac{\|M \circ (\hat{G}_{XX} - \hat{G}_{X\tilde{X}})\|_F^2}{\|\hat{G}_{XX}\|_F^2}. \tag{7}$$

Here, the symbol $\circ$ indicates element-wise multiplication, while $M = E - I \in \mathbb{R}^{p \times p}$, with E being a matrix of ones and I the identity matrix. For simplicity, the weights $\lambda_1, \lambda_2, \lambda_3$ will be set equal to one throughout this paper. The first term in (7) penalizes differences in expectation, while the second and third terms encourage the matching of the second moments. Smaller values of this loss function intuitively suggest that $\tilde{X}$ is a better second-order approximate knockoff copy of X. Since J is smooth, a second-order knockoff machine can be trained with standard techniques of stochastic gradient descent.

As we mentioned earlier, knockoffs are not uniquely defined, and it is desirable to make $\tilde{X}$ as different as possible from X. There are various ways of encouraging a machine to seek this outcome, and a practical solution inspired by [1] consists of adding a regularization term to the loss function, penalizing large pairwise empirical correlations between X and $\tilde{X}$:

$$J_{\text{decorrelation}}(\mathbf{X}, \tilde{\mathbf{X}}) = \sum_{j=1}^{p} \widehat{\text{corr}}(X_j, \tilde{X}_j). \tag{8}$$

Each term above is defined as the empirical estimate of the Pearson correlation coefficient for the jth columns of $\mathbf{X}$ and $\tilde{\mathbf{X}}$. In summary, this describes a new general procedure for sampling approximate second-order knockoffs. Compared to the original method in [1], the additional computational burden of fitting a neural network is significant. However, the tools developed in this section are valuable because they can be generalized beyond the second-order setting, as discussed next.

4.3 Higher-order machines

In order to build a general knockoff machine, one must be able to precisely quantify and control the deviation from exchangeability: the difference in distribution between $(X, \tilde{X})$ and $(X, \tilde{X})_{\text{swap}(j)}$ for each $j \in \{1, \dots, p\}$. For this purpose, we deploy the maximum mean discrepancy metric from Section 3. In order to obtain an unbiased estimate, we randomly split the data into a partition $\mathbf{X}', \mathbf{X}'' \in \mathbb{R}^{n/2 \times p}$ and define the corresponding output of the machine as $\tilde{\mathbf{X}}', \tilde{\mathbf{X}}''$. Then, it is natural to seek a machine that targets

$$\sum_{j=1}^{p} \widehat{\mathcal{D}}_{\text{MMD}} \left[(\mathbf{X}', \tilde{\mathbf{X}}'), (\mathbf{X}'', \tilde{\mathbf{X}}'')_{\text{swap}(j)} \right].$$

Above, $\widehat{\mathcal{D}}_{\text{MMD}}$ stands for the empirical estimate in (4) of the maximum mean discrepancy, evaluated with a Gaussian kernel. Intuitively, the above quantity is minimized in expectation if the knockoffs are exchangeable according to (1). This idea will be made more precise below, for a slightly modified objective function. We refer to this solution as a higher-order knockoff machine because the expansion of the Gaussian kernel into a power series leads to a characterization of (3) in terms of the distance between vectors containing all higher-moments of the two distributions [45, 47]. Therefore, our approach can be interpreted as a natural generalization of the method in [1].

Since computing $\widehat{\mathcal{D}}_{\text{MMD}}$ at each iteration may be expensive (there are p swaps), in practice we will only consider two swaps and ask the machine to minimize

$$J_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}}) = \widehat{\mathcal{D}}_{\text{MMD}} \left[(\mathbf{X}', \tilde{\mathbf{X}}'), (\tilde{\mathbf{X}}'', \mathbf{X}'') \right] + \widehat{\mathcal{D}}_{\text{MMD}} \left[(\mathbf{X}', \tilde{\mathbf{X}}'), (\mathbf{X}'', \tilde{\mathbf{X}}'')_{\text{swap}(S)} \right], \qquad (9)$$

where S indicates a uniformly chosen random subset of $\{1, \dots, p\}$ such that $j \in S$ with probability 1/2. The following result confirms that the objective function in (9) provides a sensible guideline for training knockoff machines.

Theorem 1. *Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ be a collection of independent observations drawn from P_X, and define $\tilde{\mathbf{X}}$ as the corresponding random output of a fixed machine f_θ. Then for J_{MMD} defined as in (9),*

$$\mathbb{E} \left[J_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}}) \right] \geq 0.$$

Moreover, equality holds if and only if the machine produces valid knockoffs for P_X. Above, the expectation is taken over $\mathbf{X}$, the noise in the knockoff machine, and the random swaps in the loss function.

With a finite number of observations available, stochastic gradient descent aims to minimize the expectation of (9) conditional on the data. This involves solving a high-dimensional non-convex optimization problem that is difficult to study theoretically. Nonetheless, effective algorithms exist and a weak form of convergence of stochastic gradient descent is established in Section 4.4. Therefore, these results provide a solid basis for our proposed method.

The full objective function of a knockoff machine may also include the quantities from (7) and (8), as a form of regularization, thus reading as

$$J(\mathbf{X}, \tilde{\mathbf{X}}) = \gamma J_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}}) + \lambda J_{\text{second-order}}(\mathbf{X}, \tilde{\mathbf{X}}) + \delta J_{\text{decorrelation}}(\mathbf{X}, \tilde{\mathbf{X}}). \qquad (10)$$

In the special case of $\gamma = 0$, a second-order machine is recovered, while $\delta = 0$ may lead to knockoffs with little power. The second-order penalty may appear redundant because J_{MMD} already penalizes discrepancies in the covariance matrix, as well as in all other moments. However, we have observed that setting $\lambda > 0$

```
Algorithm 1: Training a deep knockoff machine
  Input: $\mathbf{X} \in \mathbb{R}^{n \times p}$ – Training data.
         $\gamma$ – Higher-order penalty hyperparameter.
         $\lambda$ – Second-order penalty hyperparameter.
         $\delta$ – Decorrelation penalty hyperparameter.
         $\theta_1$ – Initialization values for the weights and biases of the network.
         $\mu$ – Learning rate.
         $T$ – Number of iterations.
  Output: $f_{\theta_T}$ – A knockoff machine.
  Procedure:
  for $t = 1 : T$ do
      Sample the noise realizations: $V^i \sim \mathcal{N}(0, I)$, for all $1 \leq i \leq n$;
      Randomly divide $\mathbf{X}$ into two disjoint mini-batches $\mathbf{X}', \mathbf{X}''$;
      Pick a subset of swapping indices $S \subset \{1, \dots, p\}$ uniformly at random;
      Generate the knockoffs as a deterministic function of $\theta$:
      $\tilde{X}^i = f_{\theta_t}(X^i, V^i)$, for all $1 \leq i \leq n$;
      Evaluate the objective function, using the batches and swapping indices fixed above:
      $J_{\theta_t}(\mathbf{X}, \tilde{\mathbf{X}}) = \gamma J_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}}) + \lambda J_{\text{second-order}}(\mathbf{X}, \tilde{\mathbf{X}}) + \delta J_{\text{decorrelation}}(\mathbf{X}, \tilde{\mathbf{X}})$;
      Compute the gradient of $J_{\theta_t}(\mathbf{X}, \tilde{\mathbf{X}})$, which is now a deterministic function of $\theta$;
      Update the parameters: $\theta_{t+1} = \theta_t - \mu \nabla_{\theta_t} J_{\theta_t}(\mathbf{X}, \tilde{\mathbf{X}})$;
  end
```

often helps to decrease the amount of time required to train the machine. For optimal performance, the hyperparameters should be tuned to the specific data distribution at hand. For this purpose, we discuss practical tools to measure goodness of fit later in Section 5.1. Meanwhile, for any fixed choice of $(\gamma, \lambda, \delta)$, the learning strategy is summarized in Algorithm 1.

Alternative types of knockoff machines could be based on different choices of kernel or other measures of the discrepancy between two distributions. An intuitive option would be the Kullback-Leibler divergence [48], which appears at first sight to be a natural choice. In fact, a connection has been shown in [22] between this and the worst-case inflation of the false discovery rate that may occur if the variable selection relies on inexact knockoffs. In recent years, some empirical estimators of this divergence have been proposed in the literature on deep generative models [48, 49], which could also be employed for our purposes.

4.4 Analysis of the optimization algorithm

In this section, we study the behavior of Algorithm 1, establishing a weak form of convergence. For simplicity, we focus on the machine defined by the loss function in (10) with $(\gamma, \lambda, \delta) = (1, 0, 0)$. The other cases can be treated similarly and they are omitted in the interest of space. In order to facilitate the exposition of our analysis, we begin by introducing some helpful notations. Let $\mathbf{X}'_t$ and $\mathbf{X}''_t$ denote a randomly chosen partition of the fixed training set $\mathbf{X} \in \mathbb{R}^{n \times p}$. The state of the learning algorithm at time t is fully described by $\zeta_t = (\mathbf{X}'_t, \mathbf{X}''_t, \theta_t)$, where θ_t is the current configuration of the machine parameters. Conditional on the noise realizations $\varepsilon_t = (\mathbf{V}'_t, \mathbf{V}''_t)$ and the randomly chosen set of swapping indices S_t, the objective

$$J_{\text{MMD}}(\mathbf{X}'_t, \tilde{\mathbf{X}}'_t, \mathbf{X}''_t, \tilde{\mathbf{X}}''_t, S_t),$$

is a deterministic function of θ_t since $\tilde{\mathbf{X}}'_t = f_{\theta_t}(\mathbf{X}'_t, \mathbf{V}'_t)$ and $\tilde{\mathbf{X}}''_t = f_{\theta_t}(\mathbf{X}''_t, \mathbf{V}''_t)$. Above, J_{MMD} is written with a slight, but clarifying, abuse of the notation in (9). At this point, we can also define

$$J_{\theta_t} = \mathbb{E}\left[J_{\text{MMD}}(\mathbf{X}'_t, \tilde{\mathbf{X}}'_t, \mathbf{X}''_t, \tilde{\mathbf{X}}''_t, S_t) \middle| \zeta_t \right], \tag{11}$$

with the expectation taken over the noise ε_t and the choice of S_t. Let us also define ∇J_{θ_t} as the gradient of (11) with respect to θ_t. In practice, this quantity is approximated by sampling one realization of ε_t and a set of swapping indices S_t, then computing the following unbiased estimate:

$$g_t = \nabla J_{\text{MMD}}(\mathbf{X}'_t, f_{\theta_t}(\mathbf{X}'_t, \mathbf{V}'_t), \mathbf{X}''_t, f_{\theta_t}(\mathbf{X}''_t, \mathbf{V}''_t), S_t). \tag{12}$$

Since the function is deterministic because all random variables in (12) have been observed by the algorithm, backpropagation can be used to calculate the gradient on the right-hand-side. This gradient is then used to update the machine parameters in the next step, through $\theta_{t+1} = \theta_t - \mu g_t$, where μ is the learning rate. Under standard regularity conditions, we can follow the strategy of [50] to show that the algorithm tends to approach a stationary regime. In particular, we assume the existence of a finite Lipschitz constant L such that, for all θ', θ'' and all possible values of the data batches $\mathbf{X}', \mathbf{X}''$,

$$\left\| \nabla \mathbb{E}\left[J_{\text{MMD}}(\mathbf{X}', \tilde{\mathbf{X}}', \mathbf{X}'', \tilde{\mathbf{X}}'', S) \middle| \mathbf{X}', \mathbf{X}'', \theta' \right] - \nabla \mathbb{E}\left[J_{\text{MMD}}(\mathbf{X}', \tilde{\mathbf{X}}', \mathbf{X}'', \tilde{\mathbf{X}}'', S) \middle| \mathbf{X}', \mathbf{X}'', \theta'' \right] \right\|_2 \leq L \|\theta' - \theta''\|_2,$$

and we define

$$\Delta = \frac{2}{L} \sup \left(J_{\theta_1} - J^* \right).$$

Above, J_{θ_1} indicates the expected loss (11) at the first step, conditional on the data and the initialization of θ. The supremum is taken over all possible values of the data and the initial θ. The value of J^* is defined as a uniform lower bound on J_{θ_t}. Following the result of [45] that bounds the empirical estimate of the maximum mean discrepancy from below,

$$\widehat{\mathcal{D}}_{\text{MMD}}(\mathbf{X}, \mathbf{Z}) \geq -\frac{1}{n(n-1)} \sum_{i=1}^{n} \left[k(X^i, X^i) + k(Z^i, Z^i) - k(X^i, Z^i) \right], \qquad \forall \mathbf{X}, \mathbf{Z} \in \mathbb{R}^{n \times p},$$

we can conclude that a finite value of J^* can be determined from the data.

Theorem 2. *Consider a fixed training set $\mathbf{X} \in \mathbb{R}^{n \times p}$ and adopt the notation above. Assume that the gradient estimates have uniformly bounded variance; that is,*

$$\mathbb{E}\left[\|g_t - \nabla J_{\theta_t}\|_2^2 \middle| \zeta_t \right] \leq \sigma^2, \qquad \forall t \leq T,$$

for some $\sigma^2 \in \mathbb{R}$. Then for any initial state ζ_1 of the machine and a suitable value of the constant Δ defined above,

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\|\nabla J_{\theta_t}\|_2^2 \middle| \zeta_1 \right] \leq \frac{1}{T} \frac{L\Delta}{\mu\,(2 - L\mu)} + \frac{L\sigma^2\mu}{(2 - L\mu)}.$$

In particular, choosing $\mu = \min\left\{ \frac{1}{L}, \frac{\mu_0}{\sigma\sqrt{T}} \right\}$ for some $\mu_0 > 0$ gives

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\|\nabla J_{\theta_t}\|_2^2 \middle| \zeta_1 \right] \leq \frac{L^2\Delta}{T} + \left(\mu_0 + \frac{\Delta}{\mu_0} \right) \frac{L\sigma}{\sqrt{T}}.$$

In a nutshell, Theorem 2 states that the squared norm of the gradient of the loss function (11) decreases on average as $\mathcal{O}(T^{-1/2})$ when $T \to \infty$. This can be interpreted as a weak form of convergence that is not necessarily implying that θ_t will reach a fixed point. One could also follow the strategy of [51] instead of [50] to obtain a closely related result, guaranteeing that the norm of the gradient will be small at a sufficiently large and randomly chosen stopping time. It would of course be more desirable to establish the convergence in a stronger sense, perhaps to a local minimum; however, this is difficult and we are not aware of any similar results in the literature on deep moment matching networks. It should be noted that our assumption that the gradient estimates have uniformly bounded variance is not as strong as requiring the gradients to be uniformly bounded. The work of [52] provides explicit bounds in several special instances of single and multi-layer neural networks. However, we choose not to validate this assumption in our knockoff machines for two reasons. First, it is standard in the literature [50, 51]; second, a proof would need to rely heavily on a specific architecture and loss function. In practice, we observed that a learning rate in the typical range between 0.001 and 0.01 works well.

4.5 Implementation details

The construction of deep knockoff machines allows considerable freedom in the precise form of f_θ. In general, neural networks can be implemented following a multitude of different architectures, and the final choice is often guided by the experience of the practitioners. For the purpose of this paper, we describe a structure that works well across a wide range of scenarios. However, the options are virtually limitless and we expect that more effective designs will be found for more specific problems. The first layer of the neural network in our knockoff machine takes a vector of original variables X and a p-dimensional noise vector $V \sim \mathcal{N}(0, I)$ as input. Then a collection of h latent variables is produced by taking different linear combinations of the input and applying to each a nonlinear activation function. The connections in this layer are represented in the schematics of Figure 2, where $p = 3$ and $h = 5$. The same pattern of linear and nonlinear units is repeatedly applied to the hidden variables, through K layers of width h, as shown in Figure 3a. Finally, a similarly designed output layer returns a p-dimensional vector, as depicted in Figure 3b. Following the approach of generative moment matching networks [18], we replaced the unbiased maximum mean discrepancy loss in (9) with a slightly modified version that is always positive because it performs better in practice; see [18, Section 4.3] for technical details. In order to reduce the training time, the machines are fitted by applying stochastic gradient descent with momentum as it is customary in the field.

We have observed superior performance when a modified decorrelation penalty is adopted instead of the simpler expression in (8). For this purpose, we suggest using

$$J_{\text{decorrelation}}(\mathbf{X}, \tilde{\mathbf{X}}) = \|\text{diag}(\hat{G}_{X\tilde{X}}) - 1 + s^*_{\text{SDP}}(\hat{G}_{XX})\|_2^2,$$

where $\hat{G}$ is defined as in (6) and s^*_{SDP} is the solution to the semi-definite program

$$\begin{aligned} s^*_{\text{SDP}}(\Sigma) = \underset{s\in[0,1]^p}{\arg\min} & \sum_{j=1}^{p} |1 - s_j| \,, \\ \text{s.t.} \qquad & 2\Sigma \succeq \text{diag}(s). \end{aligned}$$

The above optimization problem is the same used in [1] to minimize the pairwise correlations between the knockoffs and the original variables, in order to boost the power, for the special case of $X \sim \mathcal{N}(0, \Sigma)$. Under the Gaussian assumption, the constraint $2\Sigma \succeq \text{diag}(s) \succeq 0$ is necessary and sufficient to ensure that the joint covariance matrix of $(X, \tilde{X})$ is positive semidefinite.

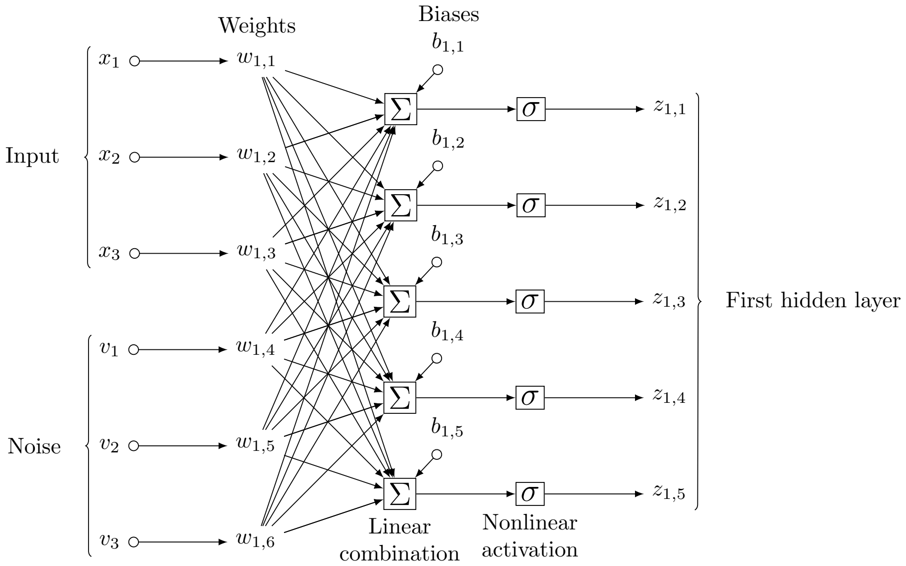
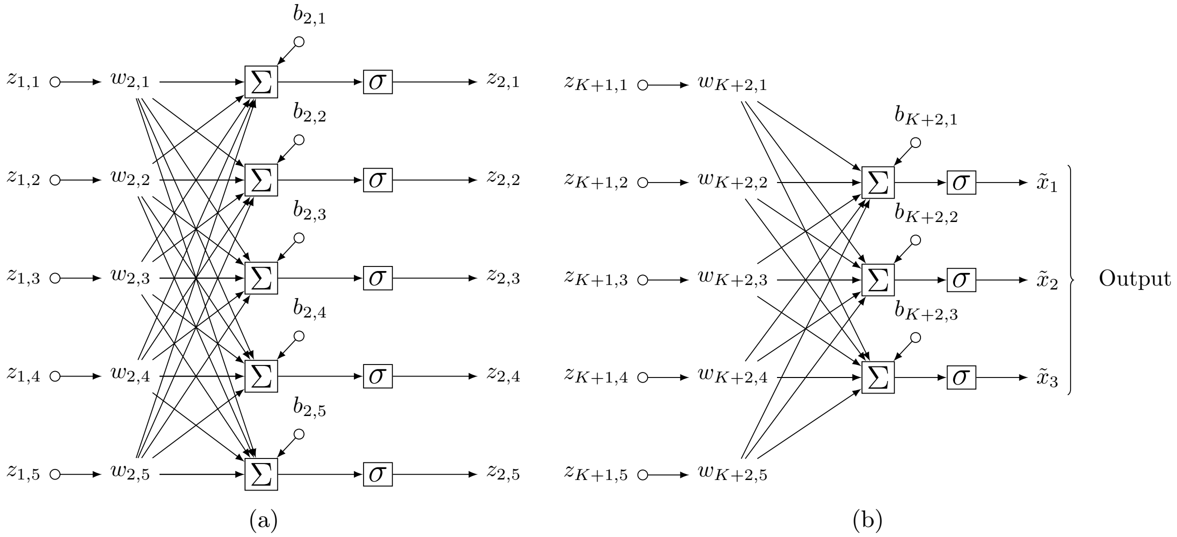


Figure 2: Connections in the input layer of a knockoff machine. This layer takes as input 3 input variables and 3 noise instances, producing 5 latent variables.



(a) (b)

Figure 3: Visualization of the connections in the hidden layers (a) and in the output layer (b) of the knockoff machine from Figure 2. The complete machine encodes a knockoff generating function $f_\theta(X, V)$ by concatenating an input layer, K hidden layers and an output layer.

5 Robustness and diagnostics

5.1 Measuring goodness-of-fit

For any fixed data source P_X, the goodness-of-fit of a conditional model producing approximate knockoff copies $\tilde{X} \mid X$ can be informally defined as the compatibility of the joint distribution of $(X, \tilde{X})$ with the exchangeability property in (1). By defining and evaluating different measures of such discrepancy, the quality of our deep knockoff machines can be quantitatively compared to that of existing alternatives. This task is a special case of the two-sample problem mentioned in Section 3, with the additional complication that a large number of distributions are to be simultaneously analyzed. In fact, for any fixed P_X and $P_{\tilde{X}|X}$, one should verify whether all of the following null hypotheses are true:

$$\mathcal{H}_0^{(j)} : P_{(X,\tilde{X})} = P_{(X,\tilde{X})_{\text{swap}(j)}}, \qquad j \in \{1, \ldots, p\}.$$

In order to reduce the number of comparisons, we will instead consider the following two hypotheses:

$$\mathcal{H}_0^{\text{full}} : P_{(X,\tilde{X})} = P_{(\tilde{X},X)}, \qquad\qquad \mathcal{H}_0^{\text{partial}} : P_{(X,\tilde{X})} = P_{(X,\tilde{X})_{\text{swap}(S)}}, \tag{13}$$

where S is a random subset of $\{1, \ldots, p\}$, chosen uniformly and independently of $X, \tilde{X}$, such that $j \in S$ with probability 1/2. Either hypothesis can be separately investigated by applying a variety of existing two-sample tests, as described below. In order to study $\mathcal{H}_0^{\text{full}}$, we define $\mathbf{Z}_1$ and $\mathbf{Z}_2$ as two independent sets of n observations, respectively drawn from the distribution of $Z_1 = (X, \tilde{X})$ and $Z_2 = (\tilde{X}, X)$. The analogous tests of $\mathcal{H}_0^{\text{partial}}$ can be performed by defining $\mathbf{Z}_2$ as the family of samples $(X, \tilde{X})_{\text{swap}(S)}$, and they are omitted in the interest of space.

Covariance diagnostics. It is natural to begin with a comparison of the covariance matrices of Z_1 and Z_2, namely $G_1, G_2 \in \mathbb{R}^{2p \times 2p}$. For this purpose, we compute the following statistic meant to test the hypothesis that $G_1 = G_2$:

$$\widehat{\varphi}_{\text{COV}} = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} \left[(Z_{1i}^\top Z_{1j})^2 + (Z_{2i}^\top Z_{2j})^2 \right] - \frac{2}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (Z_{1i}^\top Z_{2j})^2. \tag{14}$$

This quantity is an unbiased estimate of $\|G_1 - G_2\|_F^2 = \text{Tr}(G_1^\top G_1) + \text{Tr}(G_2^\top G_2) - 2\text{Tr}(G_1^\top G_2)$, if Z_1 and Z_2 have zero mean [53]. In practice, Z_1 and Z_2 will be centered if this assumption does not hold. The asymptotic distribution of (14) can be derived under mild conditions, thus yielding a non-parametric test of the null hypothesis that $G_1 = G_2$ [53]. However, since our goal is to compare knockoffs generated by alternative algorithms, we will simply interpret larger values of (14) as evidence of a worse fit.

MMD diagnostics. While being intuitive and easy to evaluate, the above diagnostic is limited as it does not capture the higher-order moments of $(X, \tilde{X})$. Therefore, different diagnostics should be used in order to have power against other alternatives. A natural choice is to rely on the maximum mean discrepancy, on which the construction of the deep knockoff machines in Section 4.3 is based. In particular, the first null hypothesis in (13) can be tested by computing

$$\widehat{\varphi}_{\text{MMD}} = \widehat{\mathcal{D}}_{\text{MMD}} \left(\mathbf{Z}_1, \mathbf{Z}_2 \right), \tag{15}$$

where the function $\widehat{\mathcal{D}}_{\text{MMD}}$ is defined as in (4). See [45] for details. Since this is an unbiased estimate of the maximum mean discrepancy between the two distributions, large values can again be interpreted as evidence against the null. On the other hand, exact knockoffs will lead to values equal to zero on average.

KNN diagnostics. The k-nearest neighbors test [42] can also be employed to obtain a non-parametric measure of goodness-of-fit. For simplicity, we consider here the special case of $k = 1$. For each sample $z_{li} \in \mathbf{Z}_l$, with $l \ \in \{1, 2\}$, we denote the nearest neighbor in Euclidean distance of z_{li} among $\mathbf{Z} = \mathbf{Z}_1 \cup \mathbf{Z}_2 \setminus \{z_{li}\}$ as $NN(z_{li})$. Then, we define $I_l(i)$ to be equal to one if $NN(z_{li}) \in \mathbf{Z}_l$ and zero otherwise, and compute

$$\widehat{\varphi}_{\text{KNN}} = \frac{1}{2n} \sum_{i=1}^{n} \left[I_1(i) + I_2(i) \right]. \tag{16}$$

This quantity is the fraction of samples whose nearest neighbor happens to originate from the same distribution. In expectation, $\widehat{\varphi}_{\text{KNN}}$ is equal to $1/2$ if the two distributions are identical, while larger values provide evidence against the null [42]. A rigorous test can be performed to determine whether to reject any given knockoff generator, by applying the asymptotic significance threshold derived in [42]. However, since exact knockoffs may be difficult to achieve in practice, we choose to use these statistics to grade the quality of different approximations. According to this criterion one should prefer knockoff constructions leading to values of this statistic that are closer to $1/2$.

Energy diagnostics. Finally, the hypotheses in (13) can also be tested in terms of the energy distance [46], defined as

$$\mathcal{D}_{\text{Energy}}(P_{Z_1}, P_{Z_2}) = 2\mathbb{E}_{Z_1,Z_2} \left[\|Z_1 - Z_2\|_2 \right] - \mathbb{E}_{Z_1,Z_1'} \left[\|Z_1 - Z_1'\|_2 \right] - \mathbb{E}_{Z_2,Z_2'} \left[\|Z_2 - Z_2'\|_2 \right], \tag{17}$$

where Z_1, Z_1', Z_2, Z_2' are independent samples drawn from P_{Z_1} and P_{Z_2}, respectively. Assuming finite second moments, one can conclude that $\mathcal{D}_{\text{Energy}} \geq 0$, with equality if and only if Z_1 and Z_2 are identically distributed [46]. Therefore, we follow the approach of [46] and define the empirical estimator

$$\widehat{\mathcal{D}}_{\text{Energy}} \left(\mathbf{Z}_1, \mathbf{Z}_2 \right) = \frac{2}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \|Z_{1i} - Z_{2j}\|_2 - \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \|Z_{1i} - Z_{1j}\|_2 - \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \|Z_{2i} - Z_{2j}\|_2,$$

and the test statistic

$$\widehat{\varphi}_{\text{Energy}} = \frac{n}{2} \widehat{\mathcal{D}}_{\text{Energy}} \left(\mathbf{Z}_1, \mathbf{Z}_2 \right). \tag{18}$$

The quantity in (18) can be shown to be always positive, and it leads to a consistent test for the equality in distribution [46], under the assumption of finite second moments. More precisely, this statistic converges in probability to $\mathbb{E} \left[\|Z_1 - Z_2\|_2 \right]$ as the sample size n grows, while diverging otherwise. Therefore, we can interpret larger values of $\widehat{\varphi}_{\text{Energy}}$ as evidence of a poorer fit.

The diagnostics defined above provide a systematic way of comparing different knockoff constructions. Sampling $\tilde{X} \mid X$ in compliance with (1) for any fixed data distribution P_X is a difficult problem. Even though the effort is motivated by the ultimate goal of performing controlled variable selection, here the challenge is greater because even roughly approximated knockoffs may sometimes happen to allow control of the rate of false positives, while failing to pass the above tests. By contrast, respecting (1) guarantees that the inference will be valid. In the experiments of Section 6 we will show that deep machines can almost match the quality of knockoffs produced by the existing specialized algorithms when prior information on P_X is known, while greatly surpassing them in other cases.

5.2 False discovery rate under model misspecification

The quality of knockoffs produced by our deep machines has been tested according to stringent measures of discrepancy with the original data. However, even when $(X, \tilde{X})$ is far from respecting the exchangeability in (1), the false discovery rate may sometimes be controlled in practice. Since a scientist aiming to perform inference on real problems cannot blindly trust any statistical method, it is important to develop a richer set of validation tools. The strategy originally proposed in [1] consists of making controlled numerical experiments that replicate the model misspecification present in the data of interest. The main idea is to sample artificial response variables Y from some known conditional likelihood given the real explanatory variables X. Meanwhile, approximate knockoff copies are generated using the best available algorithm. Since the true null hypotheses are known in this setting, the proportion of false discoveries can be computed after applying the knockoff filter. By repeating this experiment a sufficient number of times, it is possible to verify whether the false discovery rate is contained below the nominal level. These experiments help confirm whether the knockoffs can be applied because the distribution of $(X, \tilde{X})$ is the same as in the real data.

6 Numerical experiments

6.1 Experimental setup

The deep knockoff machine presented in Section 4 has been implemented in Python using the PyTorch library, following the design outlined in Section 4.5. The activation units in each layer of the neural network sketched in Figure 2 are the parametric rectified linear unit functions [54]. Between the latter and the linear combinations, an additional batch normalization function [55] is included. The width h of the hidden layers should in general depend on the dimension p of X, and the guideline $h = 10p$ works well in practice. Six such layers are interposed between the input and the output of the machine, each parametrized by different weight matrices and biases. The maximum mean discrepancy loss function is evaluated using the Gaussian mixture kernel $k(X, X') = \frac{1}{8}\sum_{i=1}^{8} \exp[-\|X - X'\|_2^2/(2\xi_i^2)]$, with $\xi = (1, 2, 4, 8, 16, 32, 64, 128)$.

The performance of the knockoff machines is analyzed in a variety of experiments for different choices of the data distribution P_X, the results of which are separately presented in the subsections below. In each case the machines are trained on synthetic data sets containing $n = 10^4$ realizations of $X \in \mathbb{R}^p$, with $p = 100$. Stochastic gradient descent is applied with mini-batches of size $n/4$ and learning rate $\mu = 0.001$, for a total number of gradients steps $T = 10^5$. A few different values of the hyperparameters in Algorithm 1 are considered, in the proximity of $(\gamma, \lambda, \delta) = (1, 1, 1)$. The performance of the deep knockoff machine is typically not very sensitive to this choice, although we will discuss how different ratios work better with certain distributions. Upon completion of training, the goodness-of-fit of the machines is quantified in terms of the metrics defined in Section 5.1, namely the matching of second moments (14), the maximum mean discrepancy score (15), the k-nearest neighbors test (16) with $k = 1$ and the energy test (18). These measures are evaluated on knockoff copies generated for 1000 previously unseen independent samples drawn from the same distribution P_X. The diagnostics obtained with deep knockoff machines are compared against those corresponding to other existing algorithms. A natural benchmark in all scenarios exposed below is the original second-order method in [1], which is applied by relying on the empirical covariance matrix $\hat{\Sigma}$ computed on the same data used to train the deep machine. Moreover, we also consider exact knockoff constructions with perfect oracle knowledge of P_X as ideal competitors.

Finally, numerical experiments are carried out by performing variable selection in a controlled setting, where the response is simulated from a known conditional likelihood. For each sample $i \in \{1, \ldots, m\}$, the response variable $Y^i \in \mathbb{R}$ is sampled according to $Y^i \sim \mathcal{N}(X^i\beta, 1)$, with $\beta \in \mathbb{R}^p$ containing 30 randomly chosen non-zero elements equal to $a/\sqrt{m}$. The experiments are repeated 1000 times, for different values of the signal amplitude a and the number of observations m. The importance measures are defined by fitting the elastic-net [56] on the augmented data matrix $[\mathbf{X}, \tilde{\mathbf{X}}] \in \mathbb{R}^{m \times 2p}$ and $\mathbf{Y} \in \mathbb{R}^m$. More precisely, we compute

$(\hat{\beta}, \tilde{\beta}) \in \mathbb{R}^{2p}$ as

$$(\hat{\beta}, \tilde{\beta}) = \underset{(b,\tilde{b})}{\arg\min} \left\{ \frac{1}{m} \| \mathbf{Y} - \mathbf{X}b - \tilde{\mathbf{X}}\tilde{b} \|_2^2 + (1-\alpha)\frac{\tau}{2} \left(\|b\|_2^2 + \|\tilde{b}\|_2^2 \right) + \alpha\tau \left(\|b\|_1 + \|\tilde{b}\|_1 \right) \right\}, \tag{19}$$

with the value of τ tuned by 10-fold cross validation and some fixed $\alpha \in [0,1]$. The knockoff filter is applied on the statistics $W_j = |\hat{\beta}_j| - |\tilde{\beta}_j|$, for all $1 \le j \le p$, at the nominal level $q = 0.1$. The power and the false discovery rate corresponding to knockoffs generated by different algorithms can be evaluated and contrasted, as a consequence of the exact knowledge of the ground truth. It is important to stress that these experiments and all the diagnostics described above only rely on new observations from P_X, generated independently of those on which the machine is trained.

6.2 Multivariate Gaussian

The first example that we present concerns the multivariate Gaussian distribution, for which the exact construction of knockoffs in [1] provides the ideal benchmark. For simplicity we consider P_X to be an autoregressive process of order one, with correlation parameter $\rho = 0.5$, such that $X \sim \mathcal{N}(0, \Sigma)$ and $\Sigma_{i,j} = \rho^{|i-j|}$. A deep knockoff machine is trained with hyperparameters $(\gamma, \lambda, \delta) = (1, 1, 1)$ and the value of its loss (10) is plotted in Figure 4 as a function of the training time.

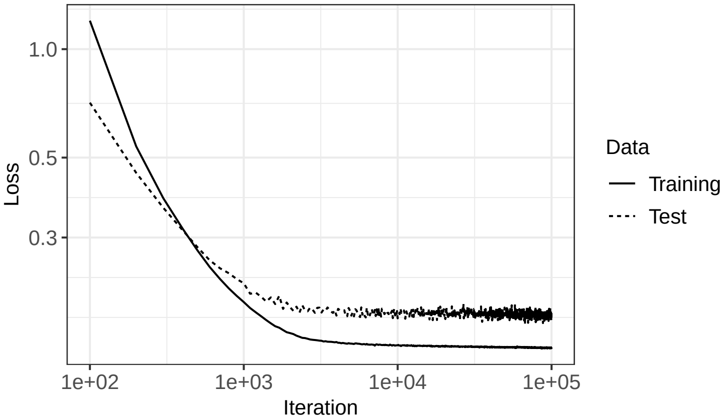
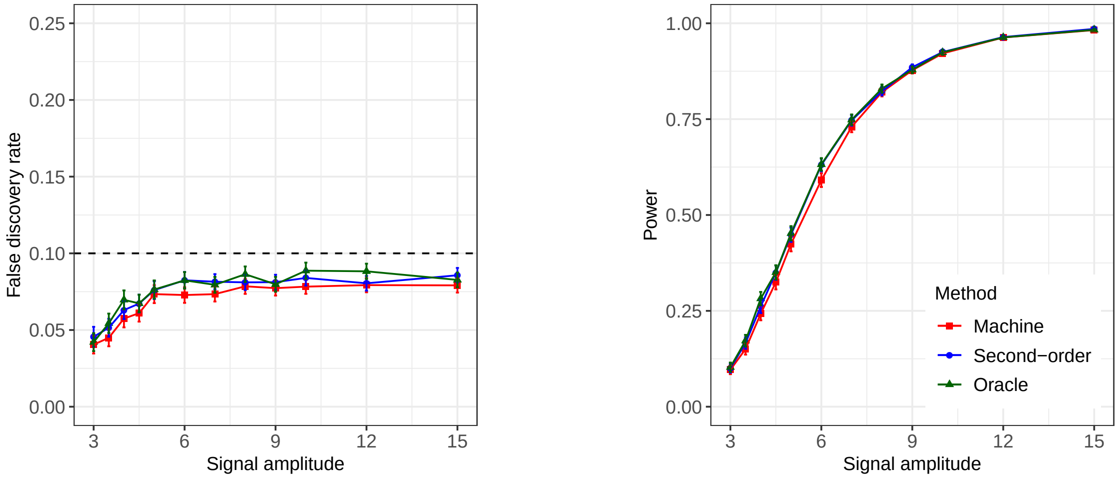


Figure 4: Evolution of the objective function for a deep machine while learning to generate knockoffs for multivariate Gaussian variables. The continuous line shows the loss (10) on the training set, while the dashed line indicates the loss evaluated on an independent test set.

The controlled numerical experiments are carried out on synthetic datasets containing $m = 150$ samples, and setting $\alpha = 0.1$ in (19). The results corresponding to the deep machine are shown in Figure 5 as a function of the signal amplitude. The performance is compared to that of the second-order method in [1] and an oracle that constructs exact knockoffs by applying the formula in (5) with the true covariance matrix Σ. The value of s in (5) is determined by solving the semi-definite program [1] from Section 4.5. The goodness-of-fit of these three alternative knockoff constructions is further investigated in terms of the diagnostics defined earlier, as shown in Figure 6. Unsurprisingly, the knockoffs generated by the oracle are perfectly exchangeable, while the deep machine and the second-order knockoffs are almost equivalent. Finally, Figure 7 suggests that the oracle has the potential to be slightly more powerful, as it can generate knockoffs with smaller pairwise correlations with their original counterparts.



(a)

(b)

Figure 5: Numerical experiments with multivariate Gaussian variables and simulated response. The performance of the machine (red) is compared to that of second-order (blue) and oracle (green) knockoffs. The false discovery rate (a) and the power (b) are computed by averaging over 1000 independent experiments. The three curves in (b) are almost indistinguishably overlapping.

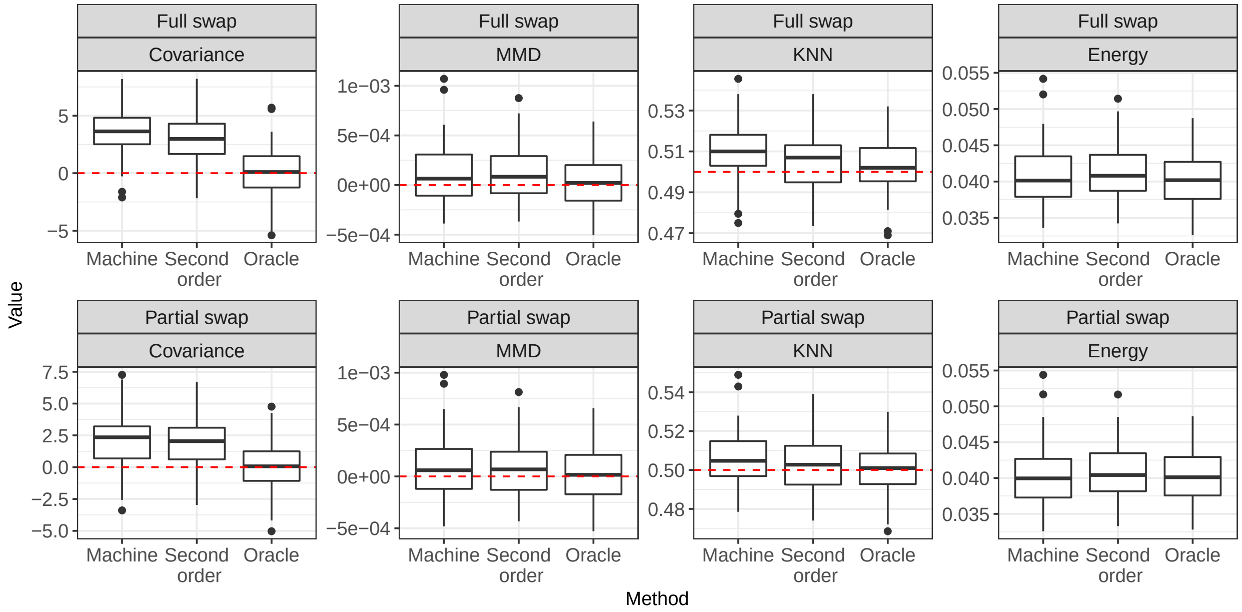


Figure 6: Boxplot comparing different knockoff goodness-of-fit diagnostics for multivariate Gaussian variables, obtained with the deep machine, the second-order method and the oracle construction, on 100 previously unseen independent datasets of size $n = 1000$.

The goodness-of-fit of the knockoff machine can also be measured against that of a misguided oracle that believes P_X to be an autoregressive process of order one with correlation parameter equal to $\bar{\rho}$. The $\tilde{X}$ thus generated are clearly not valid knockoffs unless $\bar{\rho} = \rho$. This comparison may be helpful because the limitations of the imperfect oracle are simpler to understand. For example, as $\bar{\rho}$ approaches zero, $\tilde{X}$ becomes

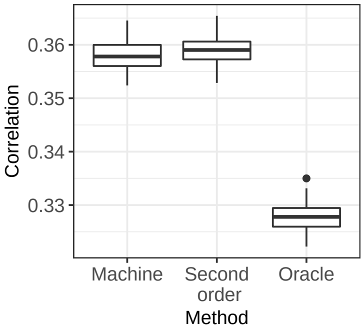


Figure 7: Boxplot comparing the average absolute pairwise correlation between variables and knockoffs for a multivariate Gaussian distribution, as in Figure 6. Lower values tend to indicate more powerful knockoffs. The numerical values on the vertical axis show that the differences between the three methods are not very large.

independent of X and the violation of (1) should be easily detectable by our tests. In the interest of space, we only compute the second-order diagnostics in (14) as a function of $\bar{\rho}$, and compare them to those in Figure 6. The results are shown in Figure 8. The misspecified oracle leads to a significantly poorer fit than the alternative methods, unless $\bar{\rho}$ is very close to ρ. This indicates that the second-order approximation and the deep machine are capturing the true P_X rather accurately, despite having very little prior information. Moreover, the experiment confirms that our diagnostics are effective at detecting invalid knockoffs.

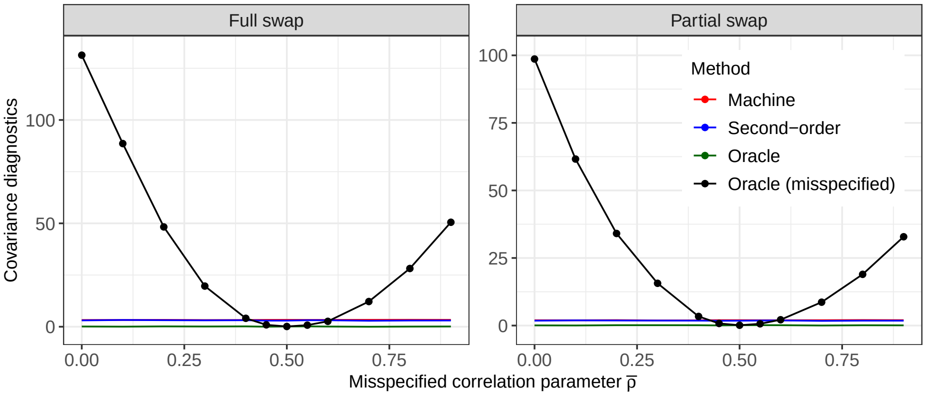


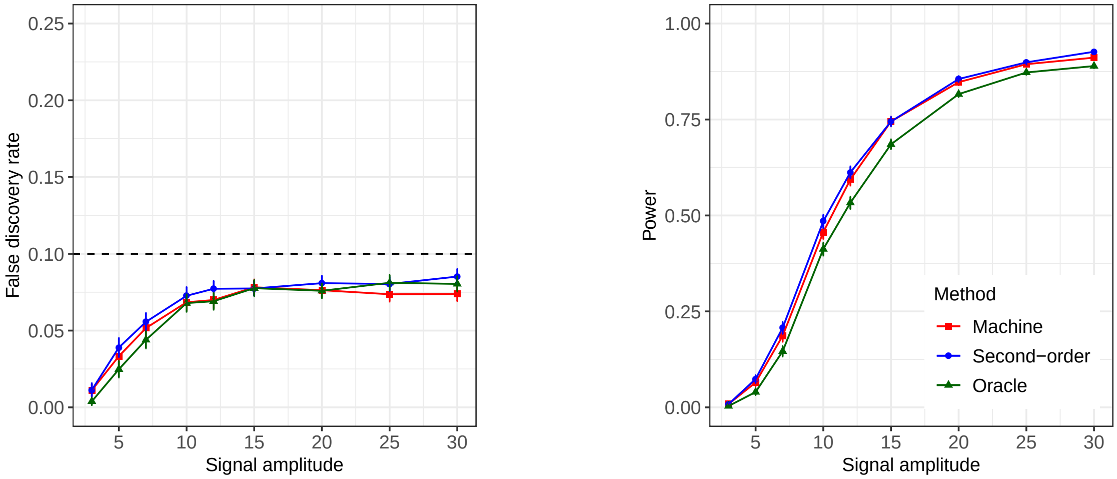
Figure 8: Covariance goodness-of-fit diagnostics for a misspecified Gaussian autoregressive knockoff oracle (black) as a function of its correlation parameter $\bar{\rho}$. These measures are compared to those of the other methods, also shown in Figure 6. The four curves indicate the expected value of the diagnostics, computed empirically on 10^6 samples. Lower values indicate a better fit and $\bar{\rho} = 0.5$ corresponds to the correct model. The lines corresponding to the deep machine and the second-order method are overlapping.

6.3 Hidden Markov model

We now consider discrete random variables $X_j \in \{0, 1, 2\}$, for $j \in \{1, \dots, p\}$, distributed according to the same hidden Markov model used in [3] to describe genotypes. In order to make the experiment more realistic, the parameters of this model are estimated from real data, by applying the imputation software fastPHASE [57] on a reference panel of 1011 individuals from the third phase of the International HapMap project, which is freely available from https://mathgen.stats.ox.ac.uk/impute/data_download_hapmap3_r2.

html. For simplicity, we restrict our attention to $p = 100$ features, corresponding to variants on chromosome one whose physical positions range between 0.758 Mb and 2.456 Mb, and whose minor allele frequency is larger than 0.1. The expectation-maximization algorithm of fastPHASE is run for 35 iterations, with the number of hidden states chosen equal to 20, and the rest of the configuration set to the default values. New observations are then sampled from the estimated P_X, so that the exact knockoff construction for hidden Markov models can be used as the oracle benchmark.

The deep knockoff machine is trained with the hyperparameters equal to $(\gamma, \lambda, \delta) = (1, 1, 1)$. The numerical experiments follow the approach outlined in Section 6.1, using $m = 150$ samples in each instance and setting $\alpha = 0.1$ in (19). The power and the false discovery rate are reported in Figure 9, and are very similar across the three methods. However, the oracle is slightly more conservative. The goodness-of-fit diagnostics in Figures 10 and 11 indicate that the machine is almost equivalent to the second-order approximation. It may be possible to improve the performance of this deep knockoff machine by changing its architecture to account for the discrete values of this data or by tuning it more carefully.



(a)

(b)

Figure 9: Numerical experiments with variables from a hidden Markov model. The other details are as in Figure 5.

6.4 Gaussian mixture model

The next example considers a multivariate Gaussian mixture model with equal proportions. In particular, we assume that each $X \in \mathbb{R}^p$ is independently sampled from

$$X \sim \begin{cases} \mathcal{N}(0, \Sigma_1), & \text{with probability } \frac{1}{3}, \\ \mathcal{N}(0, \Sigma_2), & \text{with probability } \frac{1}{3}, \\ \mathcal{N}(0, \Sigma_3), & \text{with probability } \frac{1}{3}, \end{cases}$$

where the covariance matrices Σ_1, Σ_2 and Σ_3 have the same autoregressive structure as in Section 6.2, with $\rho_1 = 0.3$, $\rho_2 = 0.5$ and $\rho_3 = 0.7$, respectively. Exact knockoffs can be constructed by applying the procedure described in [9] to the true model parameters. This oracle performs two simple steps. First, a latent mixture allocation $Z \in \{1, 2, 3\}$ is sampled from its posterior distribution given the observed X. Second, an exact multivariate Gaussian knockoff copy $\tilde{X}$ is produced conditional on X and Z. We then

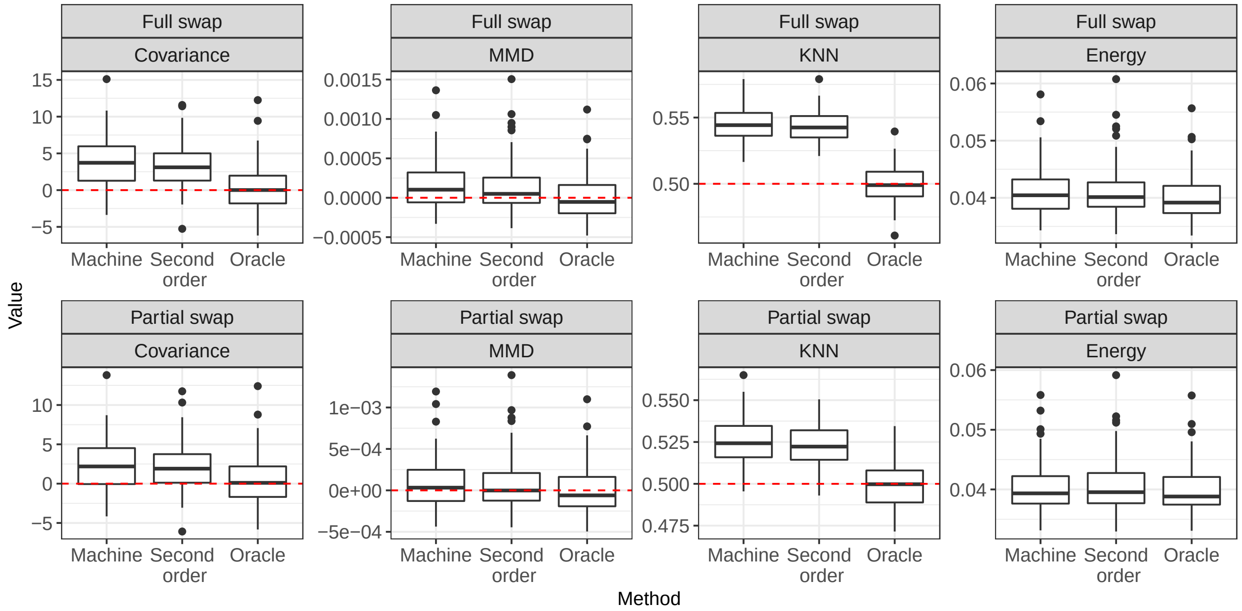


Figure 10: Boxplot comparing different knockoff diagnostics for variables sampled from a hidden Markov model. The other details are as in Figure 6.

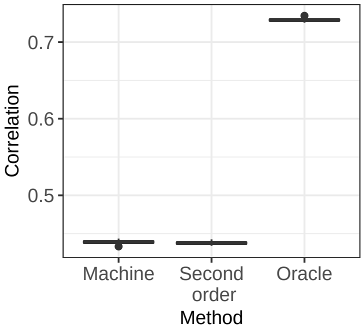


Figure 11: Boxplot comparing the average absolute pairwise correlation between variables and knockoffs for a hidden Markov model. The other details are as in Figure 7.

proceed with the experiments defined in Section 6.1, on $m = 150$ samples and setting $\alpha = 0.1$ in (19). The deep machine is trained with hyperparameters equal to $(\gamma, \lambda, \delta) = (1, 1, 1)$. The results of the numerical simulations, presented in Figure 12, show that the machine and the second-order knockoffs behave as well as the oracle. The goodness-of-fit diagnostics are reported in Figures 13 and 14. These measures indicate that the second-order method and the deep machine are essentially equivalent, while nearly as accurate as the oracle.

6.5 Multivariate Student's t-distribution

In the previous experiments, deep knockoff machines matched the performance of its best competitors. At the same time, the second-order method never failed to control the false discovery rate. In contrast, the next

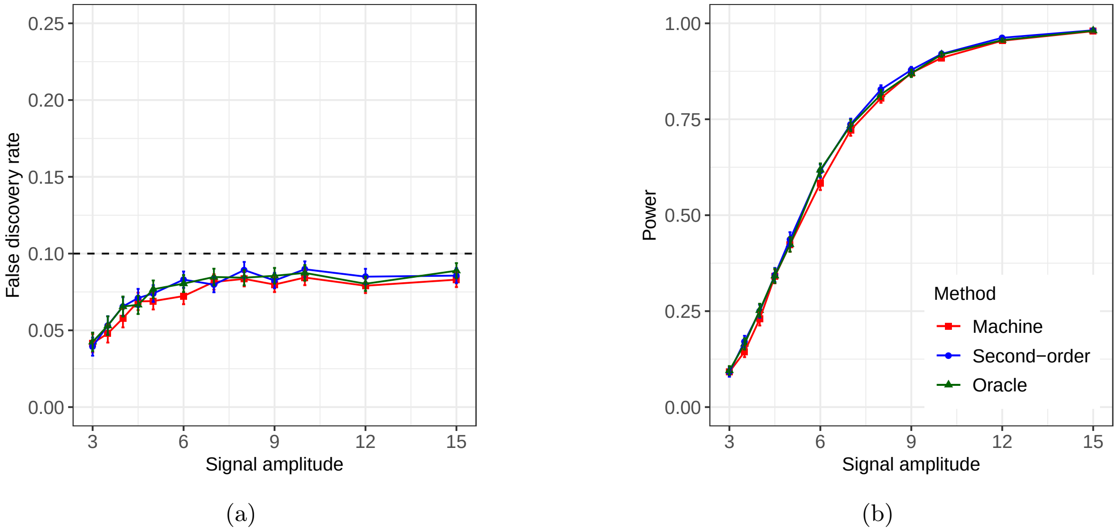


Figure 12: Numerical experiments with variables from a multivariate Gaussian mixture. The other details are as in Figure 5. The three curves in (b) are almost indistinguishably overlapping.

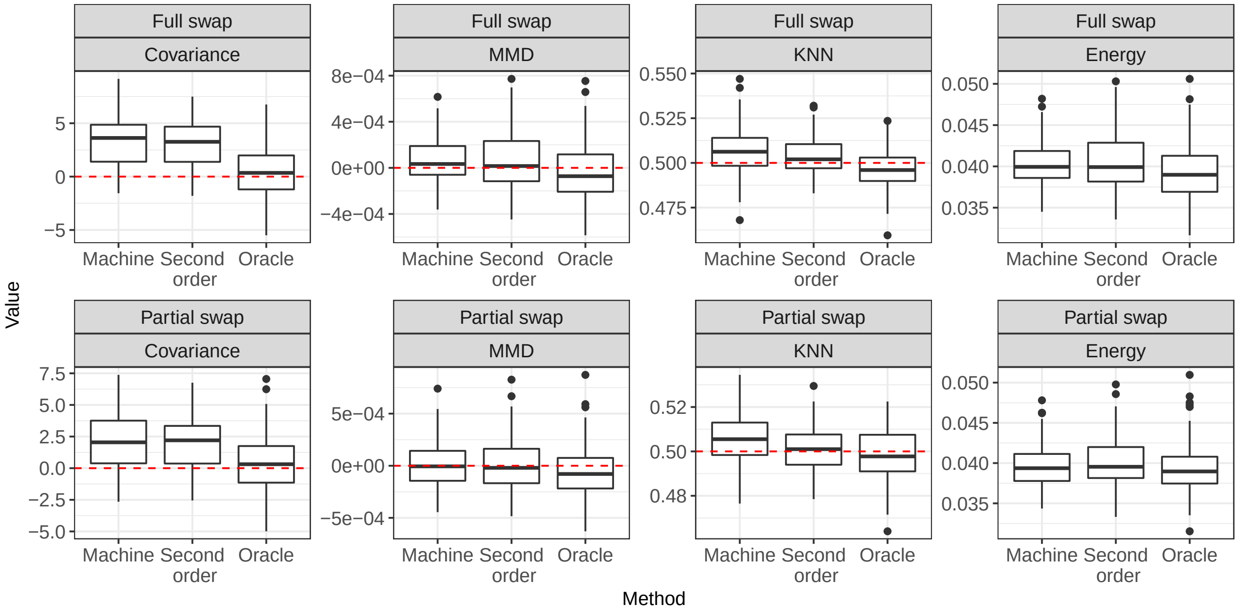


Figure 13: Boxplot comparing different knockoff diagnostics for variables sampled from a multivariate Gaussian mixture. The other details are as in Figure 6.

two examples show that second-order knockoffs can indeed fail, and quite spectacularly, in the presence of different data distributions P_X. In particular, we now consider a multivariate Student's t-distribution with zero mean and $\nu = 3$ degrees of freedom, defined such that

$$X = \sqrt{\frac{\nu - 2}{\nu}} \frac{Z}{\sqrt{\Gamma}},$$

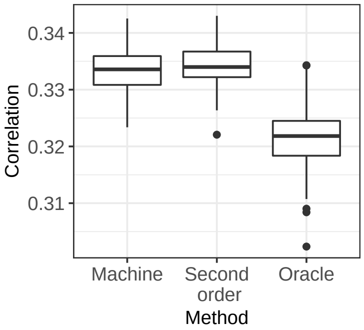


Figure 14: Boxplot comparing the average absolute pairwise correlation between variables and knockoffs for a multivariate Gaussian mixture model. The other details are as in Figure 7.

where $Z \sim \mathcal{N}(0, \Sigma)$ and Γ is independently drawn from a gamma distribution with shape and rate parameters both equal to $\nu/2$. The covariance matrix Σ is that of an autoregressive process of order one with $\rho = 0.5$, as defined in Section 6.1. The scaling factor ensures that each variable has unit variance, while their tails remain heavy. In fact, moments of order ν or higher are not finite.

The numerical experiments of Section 6.2 are carried out using $m = 200$ samples and setting $\alpha = 0$ in (19). The performance of a deep machine is only compared to that of the second-order method. An oracle for this P_X is not considered here because it is not well known, although it can be derived. The deep machine is trained with the hyperparameters $(\gamma, \lambda, \delta) = (1, 0.01, 0.01)$ because we expect that less weight should be given to the empirical covariance matrix, which is less reliable than those in the previous experiments. The results shown in Figure 15 indicate that the deep knockoffs control the false discovery rate while second-order knockoffs fail. The goodness-of-fit diagnostics are reported in Figures 16 and 17, illustrating that the deep machine significantly outperforms the second-order knockoffs.

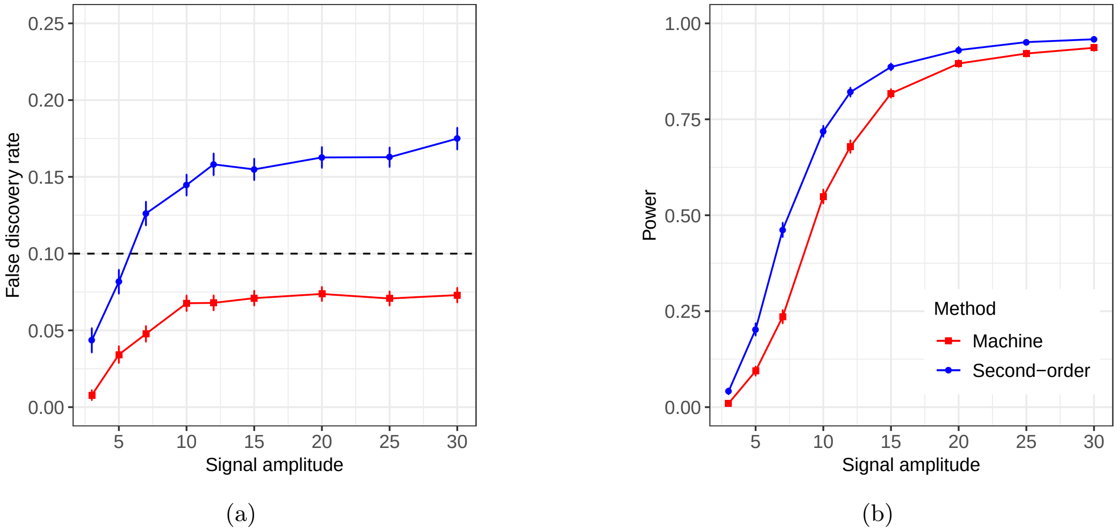


Figure 15: Numerical experiments with a multivariate Student's t-distribution. The other details are as in Figure 5.

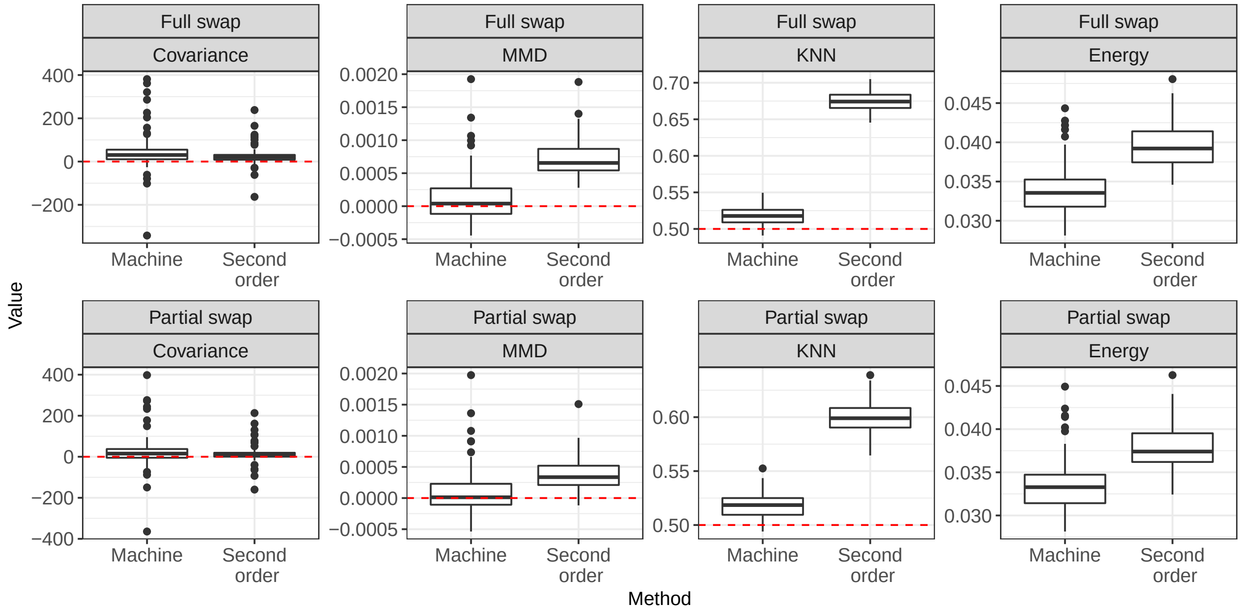


Figure 16: Boxplot comparing different knockoff diagnostics for variables sampled from a multivariate Student's t-distribution. The other details are as in Figure 6.

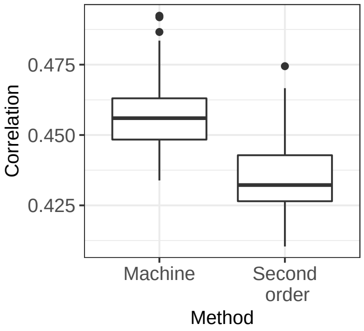


Figure 17: Boxplot comparing the average absolute pairwise correlation between variables and knockoffs for a multivariate Student's t-distribution. The other details are as in Figure 7.

6.6 Sparse Gaussian variables

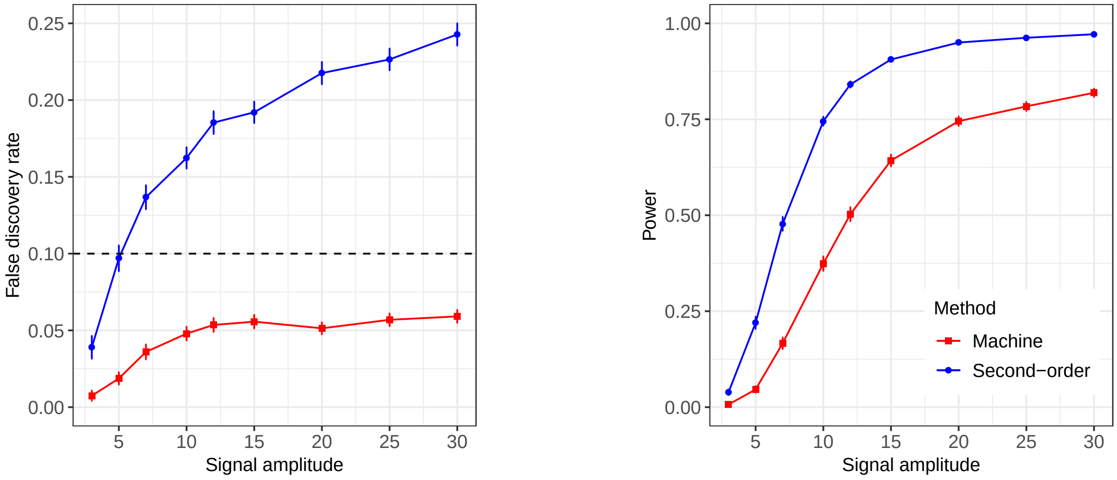
Finally, a second example is presented in which second-order knockoffs do not control the false discovery rate. The distribution considered here involves variables that are weakly correlated but highly dependent. In particular, a random variable $\eta \in \mathbb{R}$ is sampled from a standard normal distribution, while a random subset A of size $|A| = L$ is independently chosen from $\{1, \dots, p\}$. Then, for each $j \in \{1, \dots, p\}$, the value of X_j is given by

$$X_j = \sqrt{\frac{\binom{L}{p}}{\binom{L-1}{p-1}}} \cdot \begin{cases} \eta, & \text{if} \quad j \in A, \\ 0, & \text{otherwise.} \end{cases}$$

The scaling factor ensures that each variable has unit variance. In fact, the covariance matrix Σ corresponding to this choice of P_X can easily be shown to be equal to

$$\Sigma_{ij} = \begin{cases} 1, & \text{if } i = j, \\ \frac{L-1}{p-1}, & \text{otherwise.} \end{cases}$$

Here, we choose $L = 30$. Then, we perform the usual controlled numerical experiment on the deep machine trained with hyperparameters equal to $(\gamma, \lambda, \delta) = (1, 0.1, 1)$, using $m = 200$ samples and setting $\alpha = 0$ in (19). The hyperparameter $\lambda = 0.1$ decreases the weight given to the empirical covariance matrix, as in the previous experiment, while $\delta = 1$ ensures that the knockoffs are powerful. The performance of this machine is only compared to that of the second-order approximation. The results are shown in Figure 18, while the goodness-of-fit diagnostics can be found in Figures 19 and 20. We can see that the knockoffs generated by the machine are not exact; however, their approximation is more accurate than that of the second-order method. This improvement is also reflected in Figure 18, illustrating that the deep machine leads to successful control of the false discovery rate, unlike the second-order knockoffs. A combination of careful parameter tuning of the loss function, different network design and a larger training set may even further improve quality.



(a)

(b)

Figure 18: Numerical experiments with a sparse multivariate Gaussian distribution. The other details are as in Figure 5.

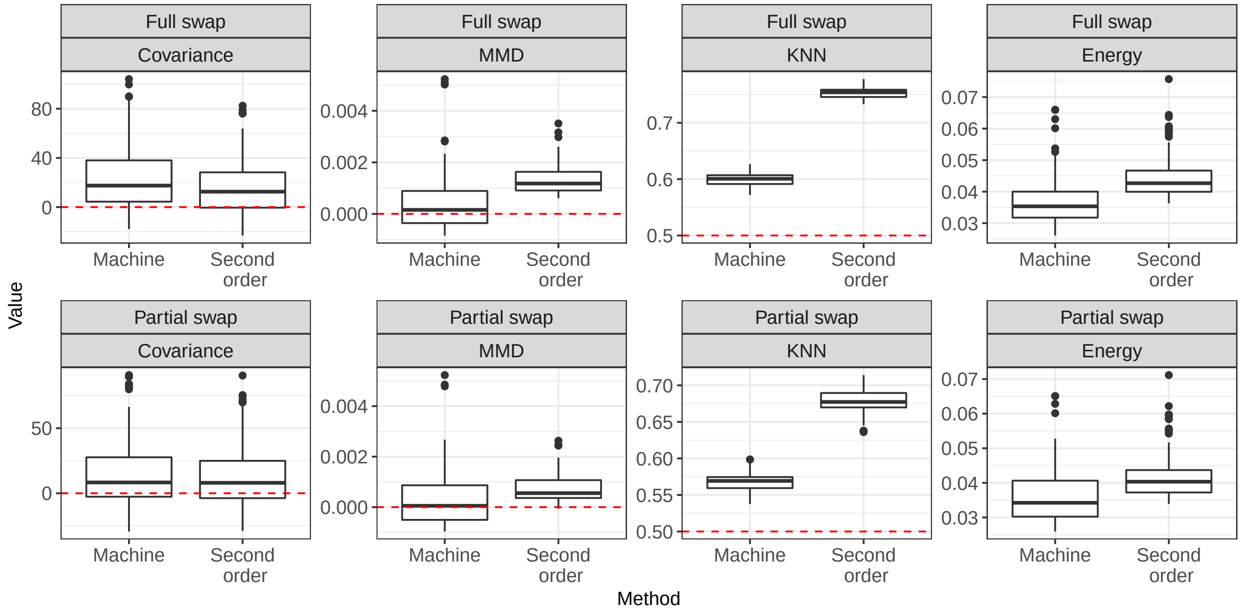


Figure 19: Boxplot comparing different knockoff diagnostics for variables sampled from a sparse multivariate Gaussian distribution. The other details are as in Figure 6.

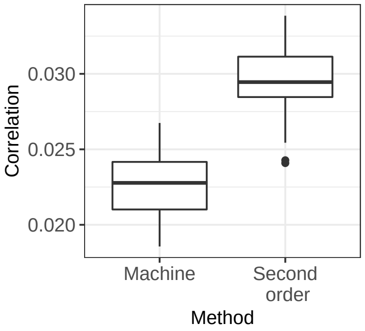


Figure 20: Boxplot comparing the average absolute pairwise correlation between variables and knockoffs for a sparse multivariate Gaussian distribution. The other details are as in Figure 7.

7 Application

7.1 Overview of the data

We deploy the deep knockoff machine to a study of variations in drug resistance among human immunodeficiency viruses of type I in order to detect important mutations [58]. We choose this application mainly for its importance and because the data are freely available from http://hivdb.stanford.edu/pages/published_analysis/genophenoPNAS2006/. Moreover, an earlier release with fewer samples also appears in the first paper on knockoffs [8]. It should be acknowledged that it is not immediately clear whether the underlying assumptions of the model-X settings are really satisfied. In particular, we do not know how realistically the samples can be described as independent and identically distributed pairs (X, Y) drawn from some joint underlying distribution. Rigorously validating these assumptions would require expert domain knowledge and additional data. Therefore, we interpret the analysis in this paper as an illustrative example of how deep knockoff machines can be used in practice, without advancing any claim of new scientific findings. In any case, it is encouraging to verify that many of the mutations discovered by our method are already known to be important, as discussed in Section 7.3 and Appendix B.

For simplicity, we focus on analyzing the resistance to one protease inhibitor drug, namely lopinavir. The response variable Y^i represents the log-fold increase in resistance measured in the ith virus. Having removed all samples containing missing values, we are left with $n = 1431$. Each of the $p = 150$ binary features X_j indicates the presence of a particular mutation. Half are chosen because they are previously known to be associated with changes in the drug resistance. The other half are chosen because they are the most frequently occurring mutations. If multiple mutations occur at the same position, the first two are treated as distinct while the others are ignored. The variables are standardized to have zero mean and unit variance, even though they have binary support. The machine in Section 6.1 is slightly modified by adding a sigmoid activation function and an affine transformation on each output node. The hyperparameters in the loss function are $(\gamma, \lambda, \delta) = (1, 1, 1)$. The machine is trained after $T = 5 \times 10^4$ gradients steps and a learning rate $\mu = 0.01$.

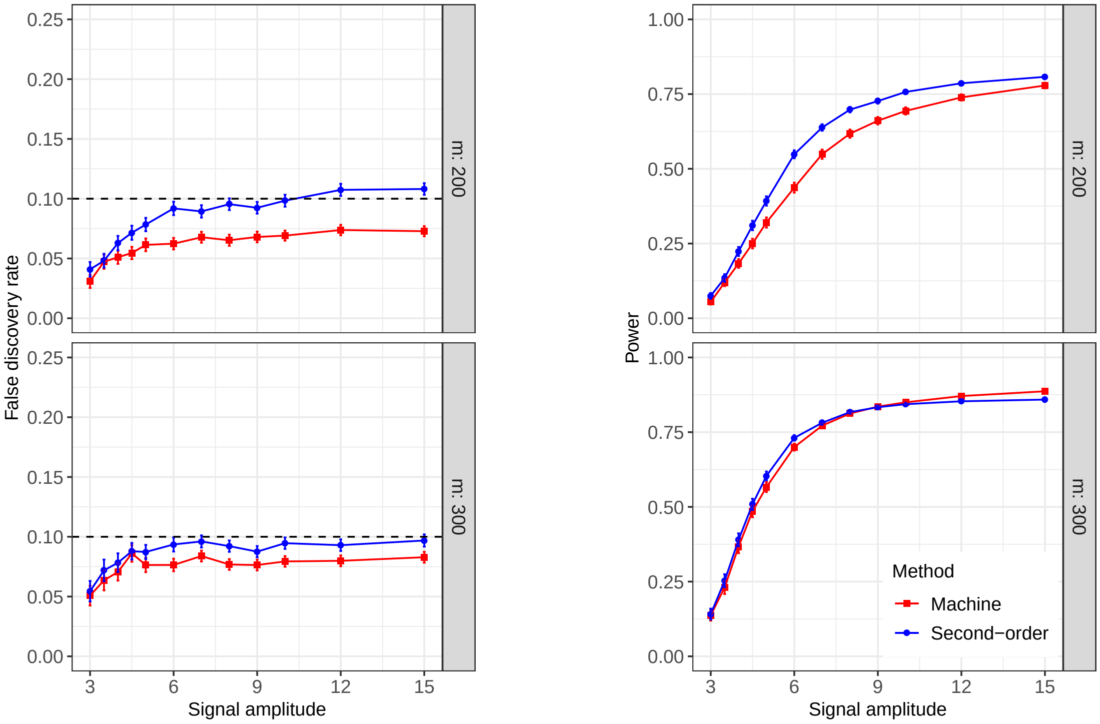
The strategy adopted for the analysis of these data is different from that described in the simulations of Section 6. A deep knockoff machine is trained on the 150 mutation features corresponding to all 1431 subjects. Since the data is limited, we fit the machine on the same samples for which we need to generate the knockoff copies to perform variable selection. Therefore, it is possible that some overfitting will occur. In other words, even though the machine thus obtained may not be very accurate on new observations of X, the knockoffs produced on the training set will be nearly indistinguishable upon a finite-sample swap with the original variables. Overfitting knockoffs has been empirically observed to lead to a loss of power at worst, while the control of the type-I errors typically remains intact [1, 3, 10]. This claim is confirmed by the results of the numerical experiments presented below, although future research should investigate a theoretical explanation of this phenomenon. For now, we accept this limitation and proceed by verifying that the machine works for our purposes.

7.2 Numerical experiments with real variables

The numerical experiments described in Section 6 are carried out using artificial response variables simulated from a known conditional linear model with 30 nonzero coefficients. Since the true population distribution of the mutations is unknown, new observations cannot be drawn from P_X. Instead, each experiment is carried out on a randomly chosen subset of size $m < n$ of the original data. Two different values of $m \in \{200, 300\}$ are considered, as discussed below. Variable selection is based on the same importance statistics of Section 6.1, setting $\alpha = 0.1$ in (19), and applying the knockoff filter to control the false discovery rate at the nominal level $q = 0.1$. As the ground truth is known, the number of true and false discoveries can be evaluated. The natural benchmark for our machine is the second-order method in [1]. The empirical covariance matrix for the latter is evaluated on the full data, in order to make a fair comparison with the deep machine. Even though the fixed-X knockoffs in [8] could in principle be used in the case where $m \geq 2p$, we have observed that they are severely underpowered in this simulation. In fact, the features exhibit strong correlations and the empirical covariance matrix in subsets of the data of size $m = 300$ is frequently singular. Therefore, fixed-X knockoffs are not ideally suited for this numerical experiment and a plot of their performance is omitted.

The results are shown in Figure 21 as a function of the signal amplitude. The deep machine successfully controls the false discovery rate, while the second-order method is slightly too liberal. Deep knockoffs often lead to more true discoveries than the second-order approximation, while making fewer mistakes. We believe that the model-X knockoff constructions applied here are overfitting, although the deep machine is more effective for variable selection within the dataset. Further insight may be provided by the goodness-of-fit diagnostics in Section 5.1; however, these would require access to additional independent samples from the same population, which we unfortunately lack. As a partial solution, one could try to split the data, even though it is not clear whether the cost would be justified, since the diagnostics will not be very powerful when evaluated on small samples.

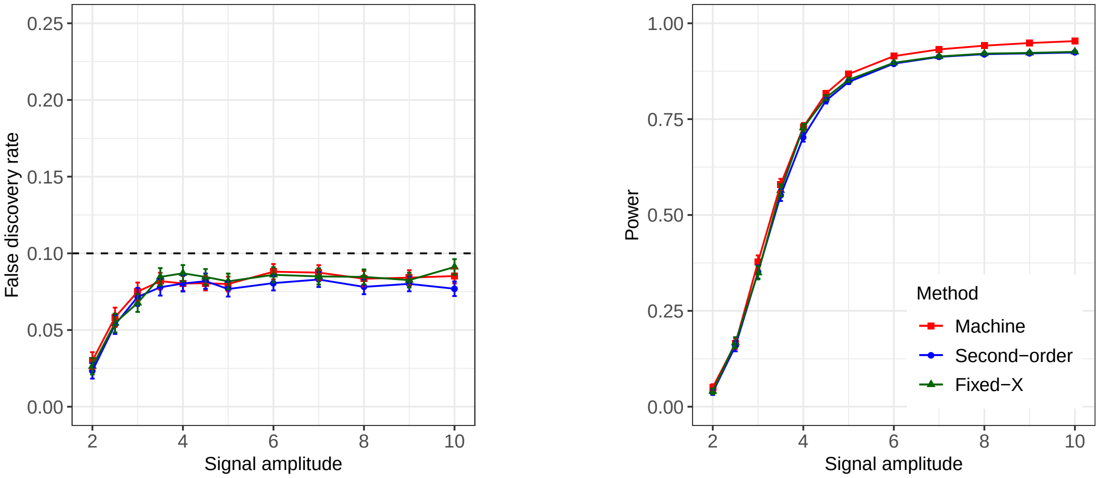
Our numerical experiments can be slightly altered to see what happens when we hold $\mathbf{X} \in \mathbb{R}^{1431 \times 150}$ constant and simulate a response variable for each observation. In theory, model-X knockoffs may not control the false discovery rate conditional on $\mathbf{X}$. However, it can be informative to apply and compare in this context the procedures described above. Since n is much greater than p and $\mathbf{X}$ is fixed, fixed-X knockoffs are a reasonable alternative to the deep machine and the second-order method. The results corresponding to the three competing approaches averaged over 1000 replications are shown in Figure 22 as a function of the signal amplitude. It is reassuring to observe that the second-order and the fixed-X knockoffs appear to control the false discovery rate and achieve similar power, while the deep machine outperforms both.



(a)

(b)

Figure 21: Numerical experiment with real human immunodeficiency virus mutation features and simulated response. The performance of the deep machine (red) is compared to that of second-order knockoffs (blue). The false discovery rate (a) and the power (b) are averaged over 1000 replications. Each replication is performed on a random subset of the original data containing $m = 200$ (top) or 300 (bottom) observations chosen without replacement.



(a)

(b)

Figure 22: Numerical experiment with real human immunodeficiency virus mutation features and simulated response. The performance of the deep machine (red) is compared to that of second-order knockoffs (blue) and fixed-X knockoff (green). The false discovery rate (a) and the power (b) are averaged over 1000 replications. Each replication is performed on the fixed original features **X**.

7.3 Results

Finally, the knockoffs generated by the machine trained in Section 7.2 are used to select important features that contribute to explaining changes in the drug resistance of the viruses. The knockoff filter is applied using the same importance statistics as above, setting $\alpha = 0.1$ in (19). The nominal false discovery rate is $q = 0.1$. In order to investigate the stability of the findings obtained with this machine, the variable selection procedure is repeated 100 times, starting from a new independent realization of the knockoffs conditional on the data. The distribution of the number of discoveries on this dataset is displayed in Figure 23, along with the analogous quantity corresponding to second-order knockoffs [1] and the randomized version of the fixed-X knockoffs [8]. The results indicate that the deep machine leads to more discoveries than the alternative approaches. This is in line with the numerical experiments presented above. It should not be surprising that fixed-X knockoffs perform better here than in Section 7.2 because the sample size is much larger. It is interesting that the selections made with our machine are quite stable upon resampling of $\tilde{X} \mid X$, unlike those of other methods. This potentially significant advantage of deep knockoff machines should be investigated more rigorously in future work. The list of discovered mutations is in large part consistent with the prior knowledge on their importance, and it is shown in Appendix B. In fact, according to the database on `https://hivdb.stanford.edu/dr-summary/comments/PI/`, many of our findings have been previously reported to have a major or accessory effect on changes in drug resistance.

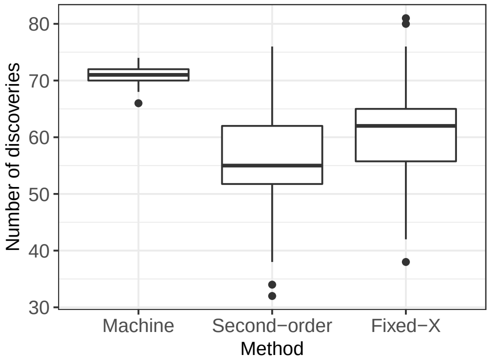


Figure 23: Boxplot of the number of drug-resistance mutations in the human immunodeficiency virus discovered using different knockoff generation methods. The variability in the results corresponds to 100 independent samples of the knockoff copies.

8 Discussion

8.1 Summary

The deep machines presented in this paper extend the knockoff method to a vast range of problems. The idea of sampling knockoff copies by matching higher moments is a natural generalization of the existing second-order approximation; however, the inherent difficulties of this approach have prompted us to exploit the powerful new methods of deep learning. The numerical experiments and the data analysis described in this paper can be reproduced on a single graphics processing unit within a few hours. We believe that the computational cost can be decreased as more experience is acquired, and applications on a larger scale should be pursued. The extensive numerical experiments show that our solution can match the performance of the available exact knockoff constructions for several data distributions, and greatly outperform the previous approximations in more complex cases. The diagnostics computed on independent test data confirm that the deep machines are correctly learning to generate valid knockoffs, without relying on prior knowledge. The theoretical results contribute to providing a principled basis for our approach. The outcomes of the data analysis are also encouraging and motivate further applications.

There is a subtle but meaningful difference between the perspective taken by the existing theory of model-X knockoffs and the common practice on real data. In principle, finite-sample control of the false discovery rate is guaranteed when the knockoff copies are constructed with respect to the true P_X. The work of [22] precisely quantifies the extent of the worst-case deviations that may occur when a fixed and misspecified distribution Q_X is used instead of P_X. However, knockoffs are often constructed using an estimated $\hat{P}_X$ obtained from the same samples used for variable selection, as discussed in Section 7. The interesting empirical observation is that when $\hat{P}_X$ overfits the training samples, knockoffs typically become more conservative rather than too liberal. To the best of our knowledge, this phenomenon is still lacking a rigorous explanation. In any case, the numerical simulations of Section 6 show that deep knockoff machines can learn how to generate valid knockoffs. In conclusion, we believe that this work is a valuable contribution because it allows the rich framework of knockoffs to be applied in very general settings. In fact, given sufficient data and adequate computing resources, deep knockoff machines can be trained on virtually any kind of features.

8.2 Our experience with other machines

This work has been primarily driven by the need to develop an effective and principled tool for the analysis of complex datasets. Generative moment matching networks are not the only technique that we considered. In fact, the worst-case perspective in the robustness theory may suggest an approach based on generative adversarial networks. We were discouraged from following that route by the complications of simultaneously fitting a generator and a discriminator. Moreover, we were mainly driven by a desire to seek a simple and practical solution, preferably building upon the well established second-order approximation. Our considerable efforts in the attempt to generate good knockoffs with a variational autoencoder could not overcome the serious limitations in power that we observed. An alternative solution was also explored, relying on deep Boltzmann machines to learn a suitably exchangeable joint distribution of $(X, \tilde{X})$ that would be consistent with the observed data. However, the computational challenges resulting from this fundamentally more difficult stochastic optimization problem eventually convinced us to search for a better path. Finally, we have found that generative moment matching networks lead to deep knockoff machines that are very effective and elegantly fit within the existing literature.

8.3 Future work

There are several paths open for future research. For example, variations of our machines could be based on different knockoff scoring functions or different regularization penalties. The deep machines described in this paper take a completely agnostic view of the data distribution, but there are many applications in which some prior knowledge of the structure of the variables is available. Exploiting it could greatly improve the computational and statistical efficiency of our method. An example arises from genome-wide association studies, where the features are naturally arranged in a sequential order and exhibit local dependencies that can be well described by a hidden Markov model. It may be interesting to develop deep knockoff machine specialized for this setting and to compare it with the procedure of [3] on a large scale. A different project could involve the extension of our toolbox of diagnostics and a systematic study of their relative strengths. An extension of the theoretical results in [22] may also be valuable. Since alternative knockoff constructions based on different deep learning techniques have been independently proposed in parallel to the writing of this paper [16, 20], it is also up to future research to extensively compare their empirical performance.

Acknowledgements

E. C. was partially supported by the Office of Naval Research (ONR) under grant N00014-16- 1-2712, by the Army Research Office (ARO) under grant W911NF-17-1-0304, by the Math + X award from the Simons Foundation and by a generous gift from TwoSigma. Y. R. was supported by the same Math + X award and ARO grant. Y. R. also thanks the Zuckerman Institute, ISEF Foundation and the Viterbi Fellowship, Technion, for supporting this research. M.S. was supported by the same Math + X award and ONR grant. We thank Stephen Bates, Nikolaos Ignatiadis and Eugene Katsevich for their insightful comments on an earlier draft of this paper.

References

[1] E. J. Candès, Y. Fan, L. Janson, and J. Lv. “Panning for gold: “model-X” knockoffs for high dimensional controlled variable selection”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 80.3 (2018), pp. 551–577.

[2] Y. Benjamini and Y. Hochberg. “Controlling the false discovery rate: a practical and powerful approach to multiple testing”. In: *Journal of the royal statistical society. Series B (Methodological)* (1995), pp. 289–300.

[3] M. Sesia, C. Sabatti, and E. J. Candès. “Gene hunting with hidden Markov model knockoffs”. In: *Biometrika* (2018), asy033.

[4] J. P. Ioannidis. “Why most published research findings are false”. In: *PLoS medicine* 2.8 (2005), e124.

[5] A. Gelman and E. Loken. “The statistical crisis in science”. In: *American scientist* 102.6 (2014), p. 460.

[6] M. Baker. “1,500 scientists lift the lid on reproducibility”. In: *Nature News* 533.7604 (2016), p. 452.

[7] M. R. Munafò et al. “A manifesto for reproducible science”. In: *Nature Human Behaviour* 1.1 (2017), p. 0021.

[8] R. F. Barber and E. J. Candès. “Controlling the false discovery rate via knockoffs”. In: *Ann. Statist.* 43.5 (2015), pp. 2055–2085.

[9] J. R. Gimenez, A. Ghorbani, and J. Zou. “Knockoffs for the mass: new feature importance statistics with false discovery guarantees”. In: *arXiv preprint arXiv:1807.06214* (2018).

[10] Y. Y. Lu, J. Lv, Y. Fan, and W. S. Noble. “DeepPINK: reproducible feature selection in deep neural networks”. In: *arXiv preprint arXiv:1809.01185* (2018).

[11] Y. Fan, J. Lv, M. Sharifvaghefi, and Y. Uematsu. “IPAD: stable interpretable forecasting with knockoffs inference”. In: *arXiv preprint arXiv:1809.05032* (2018).

[12] Z. Zheng, J. Zhou, X. Guo, and D. Li. “Recovering the graphical structures via knockoffs”. In: *Procedia Computer Science* 129 (2018), pp. 201–207.

[13] Y. Xiao et al. “Mapping the ecological networks of microbial communities from steady-state data”. In: *bioRxiv* (2017), p. 150649.

[14] Y. Xie, N. Chen, and X. Shi. “False discovery rate controlled heterogeneous treatment effect detection for online controlled experiments”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM. 2018, pp. 876–885.

[15] C. Gao et al. “Model-based and model-free machine learning techniques for diagnostic prediction and classification of clinical outcomes in Parkinson’s disease”. In: *Scientific Reports* 8.1 (2018), p. 7129.

[16] Anonymous. “KnockoffGAN: generating knockoffs for feature selection using generative adversarial networks”. In: *Submitted to International Conference on Learning Representations*. under review. 2019.

[17] M. Arjovsky and L. Bottou. “Towards principled methods for training generative adversarial networks”. In: *arXiv preprint arXiv:1701.04862* (2017).

[18] Y. Li, K. Swersky, and R. Zemel. “Generative moment matching networks”. In: *International Conference on Machine Learning*. 2015, pp. 1718–1727.

[19] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani. “Training generative neural networks via maximum mean discrepancy optimization”. In: *arXiv preprint arXiv:1505.03906* (2015).

[20] Y. Liu and C. Zheng. “Auto-encoding knockoff generator for FDR controlled variable selection”. In: *arXiv preprint arXiv:1809.10765* (2018).

[21] D. P. Kingma and M. Welling. “Auto-encoding variational Bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).

[22] R. F. Barber, E. J. Candès, and R. J. Samworth. "Robust inference with knockoffs". In: *arXiv preprint arXiv:1801.03896* (2018).

[23] L. E. Baum and T. Petrie. "Statistical inference for probabilistic functions of finite state Markov chains". In: *The annals of mathematical statistics* 37.6 (1966), pp. 1554–1563.

[24] N. M. Nasrabadi. "Pattern recognition and machine learning". In: *Journal of electronic imaging* 16.4 (2007), p. 049901.

[25] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. "A learning algorithm for Boltzmann machines". In: *Cognitive science* 9.1 (1985), pp. 147–169.

[26] Y. Burda, R. Grosse, and R. Salakhutdinov. "Importance weighted autoencoders". In: *arXiv preprint arXiv:1509.00519* (2015).

[27] C. K. Sønderby et al. "Ladder variational autoencoders". In: *Advances in neural information processing systems*. 2016, pp. 3738–3746.

[28] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. "Wasserstein auto-encoders". In: *arXiv preprint arXiv:1711.01558* (2017).

[29] I. Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[30] A. Makhzani et al. "Adversarial autoencoders". In: *arXiv preprint arXiv:1511.05644* (2015).

[31] S. Nowozin, B. Cseke, and R. Tomioka. "F-GAN: training generative neural samplers using variational divergence minimization". In: *Advances in Neural Information Processing Systems*. 2016, pp. 271–279.

[32] X. Chen et al. "Infogan: interpretable representation learning by information maximizing generative adversarial nets". In: *Advances in neural information processing systems*. 2016, pp. 2172–2180.

[33] L. Mescheder, S. Nowozin, and A. Geiger. "Adversarial variational Bayes: unifying variational autoencoders and generative adversarial networks". In: *arXiv preprint arXiv:1701.04722* (2017).

[34] M. Arjovsky, S. Chintala, and L. Bottou. "Wasserstein GAN". In: *arXiv preprint arXiv:1701.07875* (2017).

[35] T. Karras, T. Aila, S. Laine, and J. Lehtinen. "Progressive growing of gans for improved quality, stability, and variation". In: *arXiv preprint arXiv:1710.10196* (2017).

[36] C.-L. Li et al. "MMD GAN: towards deeper understanding of moment matching network". In: *Advances in Neural Information Processing Systems*. 2017, pp. 2203–2213.

[37] A. Srivastava, K. Xu, M. U. Gutmann, and C. Sutton. "Ratio matching MMD nets: low dimensional projections for effective deep generative models". In: *arXiv preprint arXiv:1806.00101* (2018).

[38] M. Arbel, D. J. Sutherland, M. Bińkowski, and A. Gretton. "On gradient regularizers for MMD GANs". In: *arXiv preprint arXiv:1805.11565* (2018).

[39] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton. "Demystifying MMD GANs". In: *arXiv preprint arXiv:1801.01401* (2018).

[40] P. J. Bickel. "A distribution free version of the Smirnov two sample test in the p-variate case". In: *The Annals of Mathematical Statistics* 40.1 (1969), pp. 1–23.

[41] J. H. Friedman and L. C. Rafsky. "Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests". In: *The Annals of Statistics* (1979), pp. 697–717.

[42] M. F. Schilling. "Multivariate two-sample tests based on nearest neighbors". In: *Journal of the American Statistical Association* 81.395 (1986), pp. 799–806.

[43] N. Henze. "A multivariate two-sample test based on the number of nearest neighbor type coincidences". In: *The Annals of Statistics* (1988), pp. 772–783.

[44] J. H. Friedman. *On multivariate goodness-of-fit and two-sample testing.* Tech. rep. Stanford Linear Accelerator Center, Menlo Park, CA (US), 2004.

[45] A. Gretton et al. "A kernel two-sample test". In: *Journal of Machine Learning Research* 13.Mar (2012), pp. 723–773.

[46] G. J. Székely and M. L. Rizzo. "Energy statistics: a class of statistics based on distances". In: *Journal of statistical planning and inference* 143.8 (2013), pp. 1249–1272.

[47] A. Cotter, J. Keshet, and N. Srebro. "Explicit approximations of the Gaussian kernel". In: *arXiv preprint arXiv:1109.4603* (2011).

[48] B. Jiang. "Approximate Bayesian computation with Kullback-Leibler divergence as data discrepancy". In: *International Conference on Artificial Intelligence and Statistics.* 2018, pp. 1711–1721.

[49] X. Nguyen, M. J. Wainwright, and M. I. Jordan. "Estimating divergence functionals and the likelihood ratio by convex risk minimization". In: *IEEE Transactions on Information Theory* 56.11 (2010), pp. 5847–5861.

[50] M. Sanjabi, J. Ba, M. Razaviyayn, and J. D. Lee. "Solving approximate Wasserstein GANs to stationarity". In: *arXiv preprint arXiv:1802.08249* (2018).

[51] S. Ghadimi and G. Lan. "Stochastic first-and zeroth-order methods for nonconvex stochastic programming". In: *SIAM Journal on Optimization* 23.4 (2013), pp. 2341–2368.

[52] V. K. Ithapu, S. N. Ravi, and V. Singh. "On architectural choices in deep learning: from network structure to gradient convergence and parameter estimation". In: *arXiv preprint arXiv:1702.08670* (2017).

[53] J. Li, S. X. Chen, et al. "Two sample tests for high-dimensional covariance matrices". In: *The Annals of Statistics* 40.2 (2012), pp. 908–940.

[54] K. He, X. Zhang, S. Ren, and J. Sun. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision.* 2015, pp. 1026–1034.

[55] S. Ioffe and C. Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *International Conference on Machine Learning.* 2015, pp. 448–456.

[56] H. Zou and T. Hastie. "Regularization and variable selection via the elastic net". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2 (2005), pp. 301–320.

[57] P. Scheet and M. Stephens. "A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase". In: *The American Journal of Human Genetics* 78.4 (2006), pp. 629–644.

[58] S.-Y. Rhee et al. "Genotypic predictors of human immunodeficiency virus type 1 drug resistance". In: *Proceedings of the National Academy of Sciences* 103.46 (2006), pp. 17355–17360.

Appendices

A Proofs

Proof of Theorem 1. Recall that J_{MMD} is defined as:

$$\mathbb{E}\left[J_{\text{MMD}}(\mathbf{X},\tilde{\mathbf{X}})\right] = \mathbb{E}\left\{\widehat{\mathcal{D}}_{\text{MMD}}\left[(\mathbf{X}',\tilde{\mathbf{X}}'),(\tilde{\mathbf{X}}'',\mathbf{X}'')\right]\right\} + \mathbb{E}\left\{\widehat{\mathcal{D}}_{\text{MMD}}\left[(\mathbf{X}',\tilde{\mathbf{X}}'),(\mathbf{X}'',\tilde{\mathbf{X}}'')_{\text{swap}(S)}\right]\right\}$$

$$= \mathbb{E}\left\{\widehat{\mathcal{D}}_{\text{MMD}}\left[(\mathbf{X}', \tilde{\mathbf{X}}'), (\tilde{\mathbf{X}}'', \mathbf{X}'')\right]\right\} + \mathbb{E}\left\{\mathbb{E}\left[\widehat{\mathcal{D}}_{\text{MMD}}\left[(\mathbf{X}', \tilde{\mathbf{X}}'), (\mathbf{X}'', \tilde{\mathbf{X}}'')_{\text{swap}(S)}\right] \mid S\right]\right\}.$$

The expectation is taken with respect to the random swap S, the data $\mathbf{X}$, its partition into $\mathbf{X}', \mathbf{X}''$ and the noise in the machine that produces $\tilde{\mathbf{X}}', \tilde{\mathbf{X}}''$. Since we know from [45] that $\widehat{\mathcal{D}}_{\text{MMD}}$ is an unbiased estimator of $\mathcal{D}_{\text{MMD}}$ and that the latter is a non-negative quantity, it follows immediately that $\mathbb{E}\left[J_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}})\right] \geq 0$. Furthermore, the samples in $\mathbf{X}$ are independent and identically distributed and the partition is randomly chosen. Therefore, it follows that

$$\begin{aligned} \mathbb{E}\left[J_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}})\right] &= \mathcal{D}_{\text{MMD}}\left[P_{(X', \tilde{X}')}, P_{(\tilde{X}'', X'')}\right] + \mathbb{E}\left\{\mathcal{D}_{\text{MMD}}\left[P_{(X', \tilde{X}')}, P_{(X'', \tilde{X}'')_{\text{swap}(S)}}\right]\right\} \\ &= \mathcal{D}_{\text{MMD}}\left[P_{(X, \tilde{X})}, P_{(\tilde{X}, X)}\right] + \mathbb{E}\left\{\mathcal{D}_{\text{MMD}}\left[P_{(X, \tilde{X})}, P_{(X, \tilde{X})_{\text{swap}(S)}}\right]\right\}. \end{aligned} \tag{20}$$

Above, the remaining expectation is taken over the random swap S. We know that the first term is equal to zero if and only if $(\tilde{X}, X)$ has the same distribution as $(X, \tilde{X})$. Moreover, if $(X, \tilde{X})_{\text{swap}(j)} \overset{d}{=} (X, \tilde{X})$ for all $j \in \{1, \dots, p\}$, it follows that $(X, \tilde{X})_{\text{swap}(S)} \overset{d}{=} (X, \tilde{X})$ for all $S \subset \{1, \dots, p\}$ (see [1]) and the second term is zero. Conversely, assuming that the second term in (20) is equal to zero implies that $(X, \tilde{X})_{\text{swap}(S)} \overset{d}{=} (X, \tilde{X})$ for all $S \subset \{1, \dots, p\}$. The last conclusion holds because any subset S has a positive probability of being chosen and the maximum mean discrepancy between any two distributions is always positive unless they are equal, in which case it vanishes.

□

Proof of Theorem 2. The strategy is inspired by Theorem 2.1 and Corollary 2.2 in [51], as well as Theorem 4.1 and Remark 4.2.1 in [50]. While the convergence result in [51] refers to a modified version of stochastic gradient descent in which the number of gradient steps is random, we consider a fixed number of steps. The approach in [50] is closer to ours.

First, it follows from the gradient update rule and a first-order expansion that

$$J_{\theta_{t+1}} \leq J_{\theta_t} + \langle \nabla J_{\theta_t}, \theta_{t+1} - \theta_t \rangle + \frac{L}{2}\mu^2 \|g_t\|_2^2.$$

Defining $\delta_t = g_t - \nabla J_{\theta_t}$, the relation $-\mu g_t = \theta_{t+1} - \theta_t$ can be used to manipulate the above inequality as in [51]:

$$\begin{aligned} J_{\theta_{t+1}} &\leq J_{\theta_t} - \mu \langle \nabla J_{\theta_t}, g_t \rangle + \frac{L}{2}\mu^2 \|g_t\|_2^2 \\ &= J_{\theta_t} - \mu \|\nabla J_{\theta_t}\|_2^2 - \mu \langle \nabla J_{\theta_t}, \delta_t \rangle + \frac{L}{2}\mu^2 \left(\|\nabla J_{\theta_t}\|_2^2 + 2\langle \nabla J_{\theta_t}, \delta_t \rangle + \|\delta_t\|_2^2\right) \\ &= J_{\theta_t} - \left(\mu - \frac{L}{2}\mu^2\right) \|\nabla J_{\theta_t}\|_2^2 - \left(\mu - L\mu^2\right) \langle \nabla J_{\theta_t}, \delta_t \rangle + \frac{L}{2}\mu^2 \|\delta_t\|_2^2. \end{aligned}$$

Summing the above inequalities over $t = 1, \dots, T$ we get

$$\begin{aligned} \left(\mu - \frac{L}{2}\mu^2\right) \sum_{t=1}^{T} \|\nabla J_{\theta_t}\|_2^2 &\leq J_{\theta_1} - J_{\theta_{T+1}} - \left(\mu - L\mu^2\right) \sum_{t=1}^{T} \langle \nabla J_{\theta_t}, \delta_t \rangle + \frac{\mu^2 L}{2} \sum_{t=1}^{T} \|\delta_t\|_2^2 \\ &\leq J_{\theta_1} - J^* - \left(\mu - L\mu^2\right) \sum_{t=1}^{T} \langle \nabla J_{\theta_t}, \delta_t \rangle + \frac{\mu^2 L}{2} \sum_{t=1}^{T} \|\delta_t\|_2^2. \end{aligned} \tag{21}$$

We now take the expectation of (21) on both sides, conditional on ζ_1. Since the estimated gradients g_t are unbiased, i.e. $\mathbb{E}\left[g_t \mid \zeta_t\right] = \nabla J_{\theta_t}$, we have

$$\begin{aligned}
\mathbb{E}\left[\langle \nabla J_{\theta_t}, \delta_t \rangle \mid \zeta_1\right] &= \mathbb{E}\left[\mathbb{E}\left[\langle \nabla J_{\theta_t}, \delta_t \rangle \mid \zeta_1, \zeta_t\right] \mid \zeta_1\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\langle \nabla J_{\theta_t}, \delta_t \rangle \mid \zeta_t\right] \mid \zeta_1\right] \\
&= \mathbb{E}\left[\langle \nabla J_{\theta_t}, \mathbb{E}\left[\delta_t \mid \zeta_t\right] \rangle \mid \zeta_1\right] \\
&= \mathbb{E}\left[\langle \nabla J_{\theta_t}, 0 \rangle \mid \zeta_1\right] = 0.
\end{aligned}$$

Substituting this result into (21), and using the assumption that $\mathbb{E}\left[\|\delta_t\|_2^2 \mid \zeta_t\right] \leq \sigma^2$, leads to

$$\left(\mu - \frac{L}{2}\mu^2\right) \sum_{t=1}^{T} \mathbb{E}\left[\|\nabla J_{\theta_t}\|_2^2 \mid \zeta_1\right] \leq J_{\theta_1} - J^* + \frac{\mu^2 L}{2} T\sigma^2.$$

Finally, multiplying both sides by $2/[LT(2\mu - L\mu^2)]$ results in

$$\begin{aligned}
\frac{1}{TL} \sum_{t=1}^{T} \mathbb{E}\left[\|\nabla J_{\theta_t}\|_2^2 \mid \zeta_1\right] &\leq \frac{2(J_{\theta_1} - J^*)}{TL\,(2\mu - L\mu^2)} + \frac{\sigma^2 \mu}{2 - L\mu} \\
&\leq \frac{\Delta}{T\,(2\mu - L\mu^2)} + \frac{\sigma^2 \mu}{2 - L\mu}.
\end{aligned}$$

The choice of $\mu = \min\left\{\frac{1}{L}, \frac{\mu_0}{\sigma\sqrt{T}}\right\}$ allows us to conclude that

$$\frac{1}{TL} \sum_{t=1}^{T} \mathbb{E}\left[\|\nabla J_{\theta_t}\|_2^2 \mid \zeta_1\right] \leq \frac{L\Delta}{T} + \left(\mu_0 + \frac{\Delta}{\mu_0}\right) \frac{\sigma}{\sqrt{T}}.$$

□

B Table of discoveries for the HIV dataset

Table 1: List of drug-resistance mutations discovered using a deep knockoff machine, compared to the findings obtained with second-order and fixed-X knockoffs. The annotation refers to existing knowledge available on the importance of each mutation.

Mutation	Annotation	Machine	Second-order	Fixed-X
46L	Major	100	100	100
47A	Major	100	100	100
47V	Major	100	100	100
48V	Major	100	100	100
50L	Major	100	100	100
50V	Major	100	100	100
54A	Major	100	100	100
54V	Major	100	100	100
76V	Major	100	100	100
82F	Major	100	100	100
82S	Major	100	100	100
82T	Major	100	100	100

84A	Major	100	100	100
54S	Major	100	99	100
54L	Major	100	98	99
82C	Major	100	97	100
54M	Major	100	96	95
48M	Major	100	90	94
32I	Major	100	82	81
48L	Major	100	73	72
30N	Major	100	56	62
48Q	Major	100	18	26
46I	Major	100	0	0
82A	Major	100	0	0
84V	Major	100	0	0
90M	Major	100	0	0
10F	Accessory	100	100	100
20T	Accessory	100	100	100
24F	Accessory	100	100	100
24I	Accessory	100	100	100
43T	Accessory	100	100	100
73C	Accessory	100	93	97
73S	Accessory	100	88	95
73T	Accessory	100	85	79
58E	Accessory	100	73	74
53L	Accessory	100	63	74
23I	Accessory	100	59	71
33F	Accessory	100	0	0
88D	Accessory	100	0	0
10V	Other	100	99	99
20I	Other	100	70	81
71I	Other	100	41	46
10I	Other	100	0	0
71V	Other	100	0	0
16A	NA	100	100	100
72M	NA	100	99	98
89I	NA	100	95	98
67F	NA	100	94	96
57K	NA	100	94	92
35N	NA	100	93	91
77I	NA	100	92	96
95F	NA	100	92	96
69K	NA	100	84	92
64L	NA	100	84	84
37D	NA	100	79	90
22V	NA	100	79	82
92K	NA	100	75	94
93L	NA	100	65	72
67E	NA	100	59	64
91S	NA	100	58	63
66V	NA	100	57	79
12A	NA	100	41	56
72R	NA	100	38	45

37C	NA	100	28	36
36I	NA	100	0	0
63P	NA	100	0	0
14R	NA	99	19	21
73A	Accessory	98	18	21
20M	Other	94	3	12
12S	NA	86	22	34
84C	Major	57	40	49
74S	Other	42	31	34
45R	NA	24	72	93
43R	NA	6	18	24