Knoop: Practical Enhancement of Knockoff with Over-Parameterization for Variable Selection

Xiaochen Zhang[1†], Yunfeng Cai[2], Haoyi Xiong[3*†]

[1]Research Center for Mathematics and Interdisciplinary Sciences, Shandong University, 266237, Qingdao, China.
[2]Yanqi Lake Beijing Institute of Mathematical Sciences And Applications, Huairou District, 100084, Beijing, China.
[3]Microsoft Corporation., Haidian District, 100085, Beijing, China.

*Corresponding author(s). E-mail(s): haoyi.xiong.fr@ieee.org;
Contributing authors: 202012079@mail.sdu.edu.cn;
yunfengcai09@gmail.com;
†These authors contributed equally to this work.

Abstract

Variable selection plays a crucial role in enhancing modeling effectiveness across diverse fields, addressing the challenges posed by high-dimensional datasets of correlated variables. This work introduces a novel approach namely *Knockoff with over-parameterization* (**Knoop**) to enhance Knockoff filters for variable selection. Specifically, **Knoop** first generates multiple knockoff variables for each original variable and integrates them with the original variables into an over-parameterized Ridgeless regression model. For each original variable, **Knoop** evaluates the coefficient distribution of its knockoffs and compares these with the original coefficients to conduct an anomaly-based significance test, ensuring robust variable selection. Extensive experiments demonstrate superior performance compared to existing methods in both simulation and real-world datasets. **Knoop** achieves a notably higher Area under the Curve (AUC) of the Receiver Operating Characteristic (ROC) Curve for effectively identifying relevant variables against the ground truth by controlled simulations, while showcasing enhanced predictive accuracy across diverse regression and classification tasks. The analytical results further backup our observations. The source codes of this work are available at https://github.com/RubyZhang166/Knoop-Knockoff-Enhancement-with-Overparametrization-for-Feature-Selection.

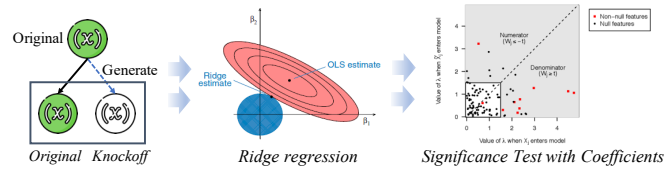
1 Introduction

Variable selection, especially supervised variable/feature selection, plays a crucial role in improving modeling effectiveness across diverse fields, such as genetics and biological science, especially with high-dimensional data exhibiting complex correlations and heterogeneity [1]. It creates a concise subset of key variables, addressing the curse of dimensionality and eliminating irrelevant and redundant variables, thereby enhancing regression model stability and prediction clarity. Variable selection strategies are categorized into model-based and model-agnostic, based on their underlying assumptions and selection methods. Common approaches include multiple testing and sparse linear models like Lasso [2, 3], ElasticNet [4], and sparse autoencoder [5]. Multiple testing assesses variable relevance through statistical conditional independence [6, 7], while Lasso uses an ℓ_1-norm penalty to reduce non-zero coefficients, stabilizing the model. The Dantzig selector [8] minimizes the residual sum of squares under an ℓ_1-norm constraint, advantageous in high-dimensional settings. ElasticNet [4], combining ℓ_1-norm and ℓ_2-norm penalties, retains variable selection properties of Lasso while handling grouping effects and multicollinearity effectively. Multiple testing can oversimplify datasets, and sparse linear models may introduce bias with highly correlated variables, motivating further research on selection techniques.

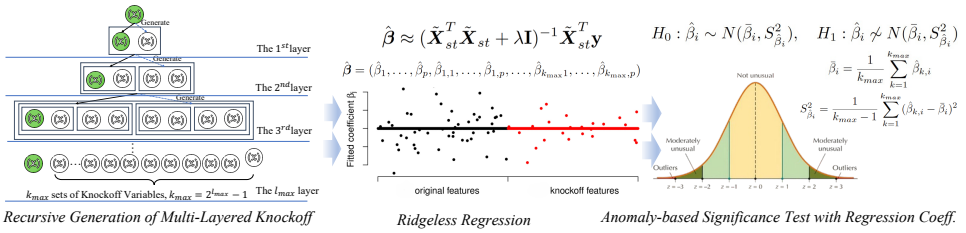
More recently, Knockoff has been introduced to statistically select variables while controlling the False Discovery Rate (FDR) [9, 10]. This approach generates synthetic "knockoff" versions of variables, comparing them with original variables to identify significant differences. Knockoff effectively handles highly correlated variables in sparse linear models, aiding in stable and unbiased variable selection [7, 11]. Its capability of managing complex models without restrictive assumptions, such as in the Model-X framework [10], allows for broad application across various fields. This adaptability makes Knockoff particularly effective in genomics and high-dimensional data settings, excelling in identifying relevant variables among many potentially correlated ones [1].

While Knockoff could outperform sparse linear models in handling highly correlated variables, it still suffers performance limitations due to the model capacity of regression models. Specifically, the vanilla Knockoff filter only generates a set of "knockoff" variables from the original ones, fits the data using the two sets of variables via linear regression model like Ridge [12] or Lasso [2], and compares the coefficient paths of every original variable and its knockoff counterpart under varying penalty to estimate their knockoff statistics. Such a simple model might be insufficient to capture complex relationships between variables and the response variable. In addition, the regularization made by Ridge or Lasso would also induce bias into the model estimation. On the other hand, the multiple Knockoff method has been proposed to run multiple rounds of Knockoff procedure and test significance of every original variable using their largest importance scores [13, 14]. Thus, it is reasonable to doubt whether the performance of Knockoff could be further improved when (1) the regression model fits the data "perfectly" [15, 16] and (2) the significance test leverages the whole distributions of knockoff and original variables' coefficients (rather than largest ones).

To address above issues, we introduce a novel Knockoff method, namely *Knockoff with Over-Parameterization* (**Knoop**), that works on top of over-parameterized



(a) Knockoff with Ridge Regression



(b) The Proposed Pipeline of **Knoop**

Fig. 1: A Brief Comparison between Knockoff and the proposed **Knoop** pipeline

linear models. As illustrated in Fig. 1, **Knoop** enhances the standard Knockoff framework by proposing a pipeline that includes three innovative components for *Knockoff variables generation*, *model regression*, and *significance testing*. Our subsequent experiments, conducted on both real-world datasets and simulations, further validated the effectiveness of this pipeline. Specifically, our work makes contribution as follows.

- As shown in Fig. 1, **Knoop** uses a *Recursive Generation of Multi-Layered Knockoff* approach to generating multi-layered knockoff variables from the original ones. In this way, the knockoff variables are expected to be independent of the response variable yet exchangeable with the original ones. Such that, every original variable and its knockoff counterpart are exchangeable and any two knockoff variables corresponding to the same original variable are also exchangeable. (Please refer to Section 3.2.2 for details.) To the best of our knowledge, this work is the first to leverage recursion to generate multiple sets of knockoff variables.
- **Knoop** incorporates both the original variables and their knockoff counterparts into a Ridgeless regression model [15] subject to the response variable, refining the coefficient estimates with enhanced fitness. Later, based on the estimates of coefficients for all variables, **Knoop** leverages an *anomaly-based significant test* to compare the coefficients of original variable against knockoff counterparts for FDR control and rank the original variables by their significance for variable selection. To the best of our knowledge, this work is the first to reformulate a significance test problem into an anomaly detection in the Knockoff framework.
- Lastly, the study conducts extensive experiments to demonstrate the superior performance of **Knoop** compared to existing methods, including sparse linear models, Knockoff and its variants, and global optimization algorithms. These experiments involve both controlled simulations and real-world datasets for classification and

regression tasks. The results highlight the effectiveness and efficiency of **Knoop** in identifying relevant variables and enhancing prediction models, showcasing its practical utility in various predictive modeling scenarios.

2 Backgrounds and Preliminaries

2.1 Variable Selection via Linear Models

Consider a dataset with n independent and identically distributed observations of p variables, which can be collectively represented in matrix form as $X \in \mathbb{R}^{n\times p}$ for the variable matrix and $\mathbf{y} \in \mathbb{R}^n$ for the response vector. An unknown projection vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top \in \mathbb{R}^p$ is assumed to represent a linear relationship as follow

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon} \text{ and } \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 I) \ .$$

The objective of variable selection is to infer the non-zero coefficients in $\boldsymbol{\beta}$, so as to understand the relationship between the variables and their responses.

2.2 Knockoff Methods

Generally, two categories of Knockoffs — fixed-X and model-X methods have been studied [9, 10]. While the fixed-X Knockoff filter generates synthetic variables with marginal correlations, independent of other variables, our work focuses on the model-X Knockoff filter generating knockoffs by modeling the joint distribution of all variables, thus managing more complex dependencies. The model-X knockoff variables $\widetilde{X}$ replicate the distribution of the original variables X [10]. These variables meet two crucial criteria: *Exchangeability*: $(X, \widetilde{X})_{\text{swap}(S)} \overset{d}{=} (X, \widetilde{X})$, achieved by switching each X_j with $\widetilde{X}_j$ in any subset S of indices and *Independence*: Ensuring that $\widetilde{X}$ is independent of the response vector $\mathbf{y}$, conditional on X. In the model-X Knockoff filter, the importance of each variable X_j is assessed through a statistic $W_j = w_j([X, \widetilde{X}], \mathbf{y})$, where w_j is a function indicating the relevancy of variables. Importantly, the sign of w_j is reversed when X_j and $\widetilde{X}_j$ are swapped within S.

$$w_j([X, \widetilde{X}]_{\text{swap}(S)}, \mathbf{y}) = \begin{cases} w_j([X, \widetilde{X}], \mathbf{y}), & j \notin S, \\ -w_j([X, \widetilde{X}], \mathbf{y}), & j \in S. \end{cases} \tag{1}$$

A threshold τ is established based on the condition as follows.

$$\tau = \min\left\{ t > 0 : \frac{1 + \#\{j : W_j \leq -t\}}{\#\{j : W_j \geq t\}} \leq q \right\}, \tag{2}$$

which guides the selection of variables for which $W_j \geq \tau$. By adhering to the above criterion, the method ensures that the rate of false discoveries among the selected variables remains within the pre-established FDR level q. To further enhance the stability of model-X Knockoff, the multiple Knockoff filter [17] has been proposed to

run multiple instances of the Knockoff procedure simultaneously and aggregating their results to control the FDR more effectively for variable selection.

Unlike existing methods, our approach **Knoop** uses a recursive approach to generating multiple sets of knockoff variables to approximate a distribution of coefficients for the knockoff, and uses both original variables and multiple knockoff sets into the regression model, enhancing both scalability and capability. Finally an anomaly-based test is proposed by **Knoop** to compare the coefficient of every single original variable with the distribution of its knockoff counterparts for significance evaluation. Such inclusion creates a larger control group of null variables, maintaining distributional similarity but ensuring independence from the response variable, thus providing a robust baseline for assessing the impact of original variables. Even compared to the multiple Knockoff method [13, 14], **Knoop** stands out in two key aspects. Firstly, its recursive generation approach ensures exchangeability between each original variable and its knockoff counterpart, as well as between any two knockoff variables corresponding to the same original variable, in a model-agnostic manner. This contrasts with the multiple Knockoff method, which relies on sequential conditional independence pairs [10] for generating multiple knockoff variables. Secondly, **Knoop** performs statistics on the entire distribution of coefficients for knockoff variables, rather than cherry-picking the ones with largest importance scores. It also reformulates the significance test as an anomaly detection problem, enabling effective FDR control.

2.3 Ridge Regression and Sparse Linear Models

When selecting variables from data, one approach is to assess the significance of the coefficients in regression models, as follows.

- *Ridge Regression [12]:* Ridge Regression integrates an ℓ_2 penalty into the loss function, effectively shrinking regression coefficients towards zero. This regularization helps mitigate overfitting by penalizing the magnitude of the coefficients. The solution is formulated as:

$$\hat{\boldsymbol{\beta}}_r = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{y} - X\beta\|_2^2 + \lambda_r \|\beta\|_2^2 \right\}, \tag{3}$$

  where λ_r is the regularization strength.
- *Lasso Regression [2]:* Lasso promotes sparsity in the coefficient vector through an ℓ_1 penalty, setting less important variable coefficients to zero, thus aiding in variable selection. The coefficients are optimized by:

$$\hat{\boldsymbol{\beta}}_l = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{y} - X\beta\|_2^2 + \lambda_l \|\beta\|_1 \right\}, \tag{4}$$

  balancing the model fit and complexity via the tuning parameter λ_l.
- *ElasticNet [4]:* Elastic Net merges the penalties of Lasso and Ridge Regression to harness the benefits of both sparsity and stability.

2.4 Ridgeless Regression and Overparameterization

Ridgeless regression extends the ordinary least squares estimator to provide a unique solution in ultra-high dimensional settings. The coefficients, $\hat{\boldsymbol{\beta}}$, can be obtained through a limiting process as $\lambda \to 0$:

$$\hat{\boldsymbol{\beta}} = \lim_{\lambda \to 0} (X^\top X + \lambda I)^{-1} X^\top \mathbf{y}, \tag{5}$$

Theoretical research has explored the conditions under which over-parameterized Ridgeless regression models attain advantages of estimation and prediction [18].

3 Methodology: Over-Parameterized Knockoff

This section presents the overall framework and core algorithms of **Knoop**.

3.1 Overall Framework Design

In this section, we describe the comprehensive design of the **Knoop** algorithm (Algorithm 1). The algorithm accepts the training set $(X, \mathbf{y})$ as input and outputs the selection result of the variables. The method comprises four principal steps: *hierarchical generation of multi-layered knockoffs, Ridgeless regression, anomaly-based significance test, and variable selection.* Each step is elaborated upon as follows.

Recursive Generation of Multi-Layered Knockoffs

Given the training set $(X, \mathbf{y})$ as input, **Knoop** employs a unique recursive generation approach to generate enhanced knockoff variable matrix X, outlined in lines 1 of Algorithm 1. This approach generates a multi-layered knockoff matrix, $\widetilde{X}$, through a recursive process, thereby increasing the model capacity with a larger number of dimensions in input features for regression. The resulting $\widetilde{X}$ is represented as:

$$\widetilde{X} = [X_1, \ldots, X_p, \widetilde{\mathbf{K}}_{1,1}, \ldots, \widetilde{\mathbf{K}}_{1,p}, \ldots, \widetilde{\mathbf{K}}_{k_{\max},1}, \ldots, \widetilde{\mathbf{K}}_{k_{\max},p}] \ .$$

Each $\widetilde{\mathbf{K}}_i = [\widetilde{\mathbf{K}}_{i,1}, \widetilde{\mathbf{K}}_{i,2} \ldots, \widetilde{\mathbf{K}}_{i,p}]$ represents a hierarchical knockoff matrix associated with X, where i ranges from 1 to $k_{\max}$. Later, to mitigate the influence of varying variable units, each column of the matrix is normalized to unit norm, resulting in a standardized multi-knockoff matrix, denoted as $\widetilde{X}_{\text{st}}$.

Ridgeless Regression

Following the above step, **Knoop** integrates both the original variables and multiple knockoff vectors into a linear regression model, utilizing a Ridgeless least squares estimator to derive the coefficient estimates, as detailed in Lines 3 of Algorithm 1. **Knoop** obtains a vector of estimated coefficients, $\hat{\boldsymbol{\beta}}$, as:

$$\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \ldots, \hat{\beta}_p, \hat{\beta}_{1,1}, \ldots, \hat{\beta}_{1,p}, \ldots, \hat{\beta}_{k_{\max},1}, \ldots, \hat{\beta}_{k_{\max},p})^\top, \tag{6}$$

Algorithm 1 `Knoop`: Knockoff with Over-Parameterization

Input data: Training data matrix $X = [x_1, \ldots, x_n]^\top$, Labels for training data $\mathbf{y}$.
Parameters: Ridgeless regression regularization parameter λ.
Output: p-values vector $\hat{\boldsymbol{P}}$ for original variables $x_1, \ldots, x_p$. Set of selected indices A within $\{1, 2, \ldots, p\}$.

1: $\widetilde{X} \leftarrow \texttt{recursKnockoff}(X)$ ▷ Generate a multi-layered knockoff matrix
2: $\widetilde{X}_{st} \leftarrow \texttt{normalize}(\widetilde{X})$ ▷ Normalize each column to unit norm
3: $\hat{\boldsymbol{\beta}} \leftarrow (\widetilde{X}_{st}^\top \widetilde{X}_{st} + \lambda I)^{-1} \widetilde{X}_{st}^\top \mathbf{y}$ ▷ Estimate the coefficients via Ridgeless regression
4: $\hat{\boldsymbol{P}} \leftarrow \texttt{estPVal}(\hat{\boldsymbol{\beta}}, \widetilde{X}_{st}, \mathbf{y})$ ▷ Estimate p-values by significance tests
5: $A \leftarrow \texttt{selectVars}(\hat{\boldsymbol{P}})$ ▷ Select top variables by p-values
6: **return** $\hat{\boldsymbol{P}}$, A, ▷ Return p-values and selected variables.

where $\hat{\beta}_1, \ldots, \hat{\beta}_p$ are the coefficients of the original variables. Each $\hat{\beta}_{k,i}$ are the coefficient for the k^{th} knockoff of the i^{th} variable, for $k = 1, \ldots, k_{\max}$ and $i = 1, \ldots, p$.

Anomaly-based Significance Test

Given the Ridgeless regression coefficients, **Knoop** estimates the p-values for the significance tests of the original variables, as executed in Lines 4 of Algorithm 1. This process involves fitting a Ridgeless regression model to the expanded training set $(\widetilde{X}_{\text{st}}, \mathbf{y})$. For a sufficiently large $k_{\max}$ (ensuring that $(k_{\max} + 1)p \geq n_{train}$), the regression model perfectly fits the training data. The estimated coefficients, denoted $\hat{\boldsymbol{\beta}}$, are then used to derive the p-values $\hat{P}_i$ for assessing variable significance. The resulting p-values $\hat{P}_i$, where smaller values indicate higher variable importance, are used to rank the variables. This step outputs a vector $\hat{\boldsymbol{P}}$, containing calculated p-values for each original variable.

Variable Selection

As was mentioned, the previous step of **Knoop** offers a significance ranking of variables based on estimated p-values. To implement `selectVars`$(\cdot)$ in Line 5 of Algorithm 1, **Knoop** provides two ways for variable selection as follows.

- *Fixed-length Selection*: In addition to variable selection by adjusted p-values (such as the Benjamini-Hochberg procedure [19]), yet another simple selection method here involves identifying the desired number of top variables with the lowest p-values. This method is particularly useful when the analysis has a pre-defined budget for the number of variables. However, when no prior information is available, determining the appropriate number of variables may necessitate model selection.
- *Varying-length Selection*: As was mentioned above, in scenarios where the optimal number of variables for selection varies, validation-based model selection becomes essential. This approach utilizes an additional validation dataset or cross-validation [20] on the training data to tune-and-error the model and determine the appropriate number of variables.

In following sections, we present the details of two core algorithms — *recursive generation of multi-layered knockoffs* and *anomaly-based significance tests*.

3.2 Recursive Generation of Multi-Layered Knockoffs

This section outlines the proposed algorithm for recursive generation of multi-layered knockoffs, detailed in Algorithm 2. Given a data matrix based on the n samples of the original p variables, denoted as X — a $n \times p$ matrix, and the number of layers desired ℓ_{max}, this step would recursively generate $2^{\ell_{max}} - 1$ sets of knockoff variables, totalling $(2^{\ell_{max}} - 1)p$ variables, from the original variables. It returns a $n \times (2^{\ell_{max}} p)$ knockoff matrix, denoted as $\widetilde{X}$, including both original variables and generated ones. Note that to simplify our analysis in the rest of our paper, we use $k_{max} = 2^{\ell_{max}} - 1$.

Algorithm 2 `recursKnockoff`: Recursive Generation of Multi-Layered Knockoffs

Input: Original variable matrix X, the number of layers desired ℓ_{max} (note that we use $k_{max} = 2^{\ell_{max}} - 1$ to simplify our analysis in the rest part of this article), regularization parameter λ.
Output: Multi-layered knockoff matrix $\widetilde{X}$.

1: $\kappa^1 \leftarrow \text{computeKnockoff}(X)$ ▷ Compute the initial knockoff matrix
2: $\mathbf{K}^1 = [X \,|\, \kappa^1]$ ▷ Concatenate X and κ^ℓ into a new matrix $\mathbf{K}^1$
3: **for** $\ell = 1$ to $\ell_{max} - 1$ **do**
4: $\quad \kappa^{\ell+1} \leftarrow \texttt{computeKnockoff}(\mathbf{K}^\ell)$ ▷ Compute the ℓ^{th}-layer knockoff matrix
5: $\quad \mathbf{K}^{\ell+1} \leftarrow [\mathbf{K}^\ell \,|\, \kappa^{\ell+1}]$ ▷ Concatenate $\mathbf{K}^\ell$ and $\kappa^{\ell+1}$ into a new matrix $\mathbf{K}^{\ell+1}$
6: **end for**
7: **return** $\widetilde{X} = \mathbf{K}^{\ell_{max}}$ ▷ Return a multi-layered (ℓ_{max} layers) knockoff matrix

3.2.1 Algorithm Design

Initially, the matrix κ^1, denoted as the matrix of knockoff counterparts of the original variable matrix X, is computed using `computeKnockoff`$(\cdot)$ and then horizontally concatenated with X to form the first layer $\mathbf{K}^1$ (lines 1-2 in Algorithm 2):

$$\mathbf{K}^1 = [X \,|\, \kappa^1]$$

Subsequent layers are constructed by repeating this process for each layer, where a new knockoff $\kappa^{\ell+1}$ is created for each existing layer $\mathbf{K}^\ell$ via `computeKnockoff`$(\cdot)$ and concatenated to form $\mathbf{K}^{\ell+1}$ (lines 3-6 in Algorithm 2):

$$\mathbf{K}^{\ell+1} = [\mathbf{K}^\ell \,|\, \kappa^{\ell+1}]$$

The procedure continues until the desired number of layers ℓ_{max} is reached, culminating in the final hierarchical knockoff matrix $\widetilde{X} = \mathbf{K}^{\ell_{max}}$. Note that, in the rest of this paper, we use a layer-column conversion and write the structure of $\widetilde{X}$ as follows.

$$\widetilde{X} = [X_1, \ldots, X_p, \widetilde{\mathbf{K}}_{1,1}, \ldots, \widetilde{\mathbf{K}}_{1,p}, \ldots, \widetilde{\mathbf{K}}_{k_{\max},1}, \ldots, \widetilde{\mathbf{K}}_{k_{\max},p}] \,,$$

where $[\widetilde{\mathbf{K}}_{2^{\ell-1},1}, \ldots, \widetilde{\mathbf{K}}_{2^{\ell-1},p}, \ldots, \widetilde{\mathbf{K}}_{2^{\ell}-1,1}, \ldots, \widetilde{\mathbf{K}}_{2^{\ell}-1,p}] = \kappa^{\ell}$ for each $\ell = 1,\ 2 \ldots, \ell_{max}$. For example, $[\widetilde{\mathbf{K}}_{1,1}, \ldots, \widetilde{\mathbf{K}}_{1,p}] = \kappa^{1}$ and $[\widetilde{\mathbf{K}}_{2,1}, \ldots, \widetilde{\mathbf{K}}_{2,p}, \widetilde{\mathbf{K}}_{3,1}, \ldots, \widetilde{\mathbf{K}}_{3,p}] = \kappa^{2}$. The overall generation $\widetilde{X}$ totals p original variables and $(2^{\ell_{max}} - 1) \cdot p$ knockoff variables (as $k_{max} = 2^{\ell_{max}} - 1$), which is then utilized for further analysis.

3.2.2 Algorithm Analysis

While traditional Knockoff methods focus on ensuring exchangeability between each original variable and its knockoff, **Knoop** is expected to maintain such exchangeability for multiple sets of knockoffs generated recursively from the same original variable. Here, we provide a brief analysis on the exchangeability of generated variables to the original ones [21, 22]. To establish our analysis, we first make the following assumptions.

A1 We adopt the knockoff generation procedure described in Model-X Knockoff filters [10]. Specifically, for any input matrix X_{inp} and its generated knockoff matrix $X_{gen} = \texttt{computeKnockoff}(X_{inp})$, in addition to the exchangeability between each column in X_{inp} and its counterpart in X_{gen}, we assume that each column in X_{inp} and X_{gen} is distinct.

A2 We assume the *transitivity* of exchangeability among multiple random matrices. Let denote V_1, V_2, and V_3 as three independent random matrices. Suppose V_1 and V_2 exchangeable and V_2 and V_3 are exchangeable. We say V_1 and V_3 should be exchangeable.

The assumption is critical to ensure the *recursive generation of multi-layered knockoff* approach would not create knockoff variables that are identical to its original variables or other knockoff ones. In our experiments, we find this assumption holds in both simulation studies or evaluations based on real-world datasets. Based on the above assumptions, we derive the main analysis result for the *recursive generation of multi-layered knockoffs* as following proposition.

Proposition 1 (Exchangability between multi-layered knockoffs)**.** *Given the input matrix* $X = [X_1, \ldots, X_p]$*, Algorithm 2 outputs a matrix* $\widetilde{X}$ *with the structure* $\widetilde{X} = [X_1, \ldots, X_p, \widetilde{\mathbf{K}}_{1,1}, \ldots, \widetilde{\mathbf{K}}_{1,p}, \ldots, \widetilde{\mathbf{K}}_{k_{max},1}, \ldots, \widetilde{\mathbf{K}}_{k_{max},p}]$*. We say that for each* $j = 1,\ 2, \ldots, k_{max}$*, each* $j' = 1,\ 2, \ldots, k_{max}$ *and* $j \neq j'$*, the matrix* X *and the sub-matrix* $\widetilde{\mathbf{K}}_j = [\widetilde{\mathbf{K}}_{j,1} \ldots \widetilde{\mathbf{K}}_{j,p}]$ *of* $\widetilde{X}$ *are exchangeable, such that*

$$[X_1, \ldots, X_p, \widetilde{\mathbf{K}}_{j,1} \ldots \widetilde{\mathbf{K}}_{j,p}]_{\text{swap}(S)} \overset{d}{=} [X_1, \ldots, X_p, \widetilde{\mathbf{K}}_{j,1} \ldots \widetilde{\mathbf{K}}_{j,p}], \tag{7}$$

and the sub-matrices $[\widetilde{\mathbf{K}}_{j,1}, \ldots, \widetilde{\mathbf{K}}_{j,p}]$ *and* $[\widetilde{\mathbf{K}}_{j',1} \ldots, \widetilde{\mathbf{K}}_{j',p}]$ *are exchangeable, such that*

$$[\widetilde{\mathbf{K}}_{j,1} \ldots \widetilde{\mathbf{K}}_{j,p}, \widetilde{\mathbf{K}}_{j',1} \ldots \widetilde{\mathbf{K}}_{j',p}]_{\text{swap}(S)} \overset{d}{=} [\widetilde{\mathbf{K}}_{j,1} \ldots \widetilde{\mathbf{K}}_{j,p}, \widetilde{\mathbf{K}}_{j',1} \ldots \widetilde{\mathbf{K}}_{j',p}], \tag{8}$$

where the operator $A_{\text{swap}(S)} \overset{d}{=} A$ *for a matrix* A *was defined in the* ***Definition 2*** *in [10] representing the distributional invariance of the matrix after certain column-wise swapping — a cornerstone of exchangeability.*

Proof. We provide a brief proof based on following steps:

Step 1. By the exchangeability property in the definition of model-X knockoff given by Definition 2 in [10], for the input matrix $X = [X_1, \ldots, X_p]$ and its knockoff matrix $\kappa^1 = [\widetilde{\mathbf{K}}_{1,1}, \ldots, \widetilde{\mathbf{K}}_{1,p}] = \texttt{computeKnockoff}(X)$, we have: $[X_1, \ldots, X_p, \widetilde{\mathbf{K}}_{1,1} \ldots \widetilde{\mathbf{K}}_{1,p}]_{\text{swap}(S)} \overset{d}{=} [X_1, \ldots, X_p, \widetilde{\mathbf{K}}_{1,1} \ldots \widetilde{\mathbf{K}}_{1,p}]$.

Step 2. Then we prove that the second layer of $\widetilde{X}$ matrix, $\widetilde{\mathbf{K}}^2$, satisfies Proposition 1. By definition of knockoff,

$$\kappa^2 \overset{d}{=} \widetilde{\mathbf{K}}^1 \tag{9}$$

Therefore, $X \overset{d}{=} [\widetilde{\mathbf{K}}_{2,1}, \ldots, \widetilde{\mathbf{K}}_{2,p}]$, and $[\widetilde{\mathbf{K}}_{1,1}, \ldots, \widetilde{\mathbf{K}}_{1,p}] \overset{d}{=} [\widetilde{\mathbf{K}}_{3,1}, \ldots, \widetilde{\mathbf{K}}_{3,p}]$ so that X and $[\widetilde{\mathbf{K}}_{2,1}, \ldots, \widetilde{\mathbf{K}}_{2,p}]$ satisfies Equation 7, $[\widetilde{\mathbf{K}}_{1,1}, \ldots, \widetilde{\mathbf{K}}_{1,p}]$ and $[\widetilde{\mathbf{K}}_{3,1}, \ldots, \widetilde{\mathbf{K}}_{3,p}]$ satisfies Equation 8. Besides, $[\widetilde{\mathbf{K}}_{2,1}, \ldots, \widetilde{\mathbf{K}}_{2,p}]$ and $[\widetilde{\mathbf{K}}_{3,1}, \ldots, \widetilde{\mathbf{K}}_{3,p}]$ satisfies Equation 8. So by Lemma *A2*, X and $[\widetilde{\mathbf{K}}_{3,1}, \ldots, \widetilde{\mathbf{K}}_{3,p}]$ satisfies Equation 7 and $[\widetilde{\mathbf{K}}_{1,1}, \ldots, \widetilde{\mathbf{K}}_{1,p}]$ and $[\widetilde{\mathbf{K}}_{2,1}, \ldots, \widetilde{\mathbf{K}}_{2,p}]$ satisfies Equation 8.

Step 3. Suppose for X and any two knockoff submatrices $D = [\widetilde{\mathbf{K}}_{j_1,1}, \ldots, \widetilde{\mathbf{K}}_{j_1,p}]$ and $E = [\widetilde{\mathbf{K}}_{j_2,1}, \ldots, \widetilde{\mathbf{K}}_{j_2,p}]$ (where $j_1 \neq j_2$) from the ℓ^{th} layer, Equation 7 and Equation 8 hold:

$$[X \quad D]_{\text{swap}(S)} \overset{d}{=} [X \quad D] \quad \text{and} \quad [D \quad E]_{\text{swap}(S)} \overset{d}{=} [D \quad E] \tag{10}$$

We now want to show that the $(\ell+1)^{th}$ layer, $\widetilde{\mathbf{K}}^{\ell+1}$, satisfies exchangeability. With Equation 10, we only need to prove that for any two knockoff submatrices in $\widetilde{\mathbf{K}}^{\ell+1}$, denoted as $F = [\widetilde{\mathbf{K}}_{j_3,1}, \ldots, \widetilde{\mathbf{K}}_{j_3,p}]$ and $G = [\widetilde{\mathbf{K}}_{j_4,1}, \ldots, \widetilde{\mathbf{K}}_{j_4,p}]$, we have: $[X \quad G]_{\text{swap}(S)} \overset{d}{=} [X \quad G]$, $[F \quad G]_{\text{swap}(S)} \overset{d}{=} [F \quad G]$, $[D \quad G]_{\text{swap}(S)} \overset{d}{=} [D \quad G]$. To prove that, note

$$\widetilde{\mathbf{K}}^{\ell} \overset{d}{=} \kappa^{\ell+1}, \quad \text{then} \quad X \overset{d}{=} [\widetilde{\mathbf{K}}_{2^{\ell-1},1}, \ldots, \widetilde{\mathbf{K}}_{2^{\ell-1},p}]. \tag{11}$$

so $[F \quad G]_{\text{swap}(S)} \overset{d}{=} [F \quad G]$. Since F can be any submatrix in $\kappa^{\ell+1}$, $[\widetilde{\mathbf{K}}_{2^{\ell-1},1}, \ldots, \widetilde{\mathbf{K}}_{2^{\ell-1},p}, G]_{\text{swap}(S)} \overset{d}{=} [\widetilde{\mathbf{K}}_{2^{\ell-1},1}, \ldots, \widetilde{\mathbf{K}}_{2^{\ell-1},p}, G]$. Then by Equation 11 and *A2*, $[X \quad G]_{\text{swap}(S)} \overset{d}{=} [X \quad G]$. And by Equation 10 and *A2*, $[D \quad G]_{\text{swap}(S)} \overset{d}{=} [D \quad G]$. Thus Proposition 1 holds for the $(\ell+1)^{th}$ layer.

Therefore, the multi-layered knockoffs generated by the recursive process maintain the exchangeability property with the original variables and among themselves. □

Remark on Proposition *1*. **Knoop** ensures the exchangeability both between each original variable and its knockoff counterparts (in Equation 7) and between any two knockoff variables rooted on the same original variable (in Equation 8).

3.3 Anomaly-based Significance Test

This section outlines the procedure of *anomaly-based significance test* based on the coefficients of Ridgeless regression. Specifically, **Knoop** considers an original variable significant (relative to the response variable) when its coefficient becomes an outlier in the total $k_{max}+1$ coefficients of the original variable together with its k_{max} knockoff counterparts. To the end, the proposed method leverages the coefficient differences between the original variables and their knockoff counterparts to estimate the p-values, prioritizes variables by their significance, and outputs a ranking list of variables for selection purposes.

Algorithm 3 `estPVal`: Anomaly-based Significance Test

Input: Regression coefficients $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \ldots, \hat{\beta}_p, \hat{\beta}_{1,1}, \ldots, \hat{\beta}_{1,p}, \ldots, \hat{\beta}_{k_{max},1}, \ldots, \hat{\beta}_{k_{max},p})^\top$, where $\hat{\beta}_i$ represents the coefficient for the i^{th} original variable and $\hat{\beta}_{k,i}$ represents the coefficient for the k^{th} knockoff of the i^{th} variable.
Output: p-values vector $\hat{\boldsymbol{P}}$ for original variables $x_1, \ldots, x_p$.

1: **for** $i = 1, \ldots, p$ **do**
2: $\quad \bar{\beta}_i \leftarrow \sum_{k=1}^{k_{max}} \hat{\beta}_{k,i}/k_{max}$
3: $\quad S_{\hat{\beta}_i} \leftarrow \sqrt{(\sum_{k=1}^{k_{max}} (\hat{\beta}_{k,i} - \bar{\beta}_i)^2)/(k_{max}-1)}$ ▷ Compute the sample mean $\bar{\beta}_i$ and standard deviation $S_{\hat{\beta}_i}$ of all k_{max} knockoff coefficients for the i^{th} variable
4: $\quad \hat{Z}_i \leftarrow \frac{\bar{\beta}_i - \hat{\beta}_i}{S_{\hat{\beta}_i}}$ ▷ Compute the Z-statistic by coefficient differences
5: $\quad \hat{P}_i \leftarrow 2\Phi(-\left|\hat{Z}_i\right|)$ ▷ Calculate the two-tailed p-value using the Z-statistic
6: **end for**
7: **return** $\hat{\boldsymbol{P}}$ ▷ The scalar $\hat{P}_i$ in $\hat{\boldsymbol{P}}$ is the p-value for the i^{th} variable.

3.3.1 Coefficients Distribution and Significance Test

Given that the k_{max} knockoff counterparts of each original variable are independent and identically distributed, we make assumptions as follows.

A3 This work follows existing studies on the distribution of regression coefficients [23–25] and assumes that for any original variable, the regression coefficients of its knockoff counterparts are normally distributed. Such that $\hat{\beta}_{k,i} \sim N(\mu_i, \sigma_i^2)$, where μ_i and σ_i^2 refer to the mean and variance of the coefficients distribution of knockoff variables for the i^{th} original variable, for each $k = 1, 2, 3, \ldots, k_{max}$.

A4 Upon the above assumption, this work further makes assumption on the test of independence. Such that, when the coefficients of an original variable x_i and its knockoff counterparts follows the same distribution, the original variable x_i is considered to be independent of the response $\mathbf{y}$. As we cannot distinguish the original variable from its knockoff counterparts using their coefficients and knockoffs were generated without seeing $\mathbf{y}$.

Hereby, we establish the *anomaly-based significance test*, using the sample mean and variance estimated from the coefficients $\hat{\beta}_{k,i}$ for $k = 1, 2, 3, \ldots, k_{max}$, as follows.

Definition 1 (Anomaly-based Significance Test)**.** *For the i^{th} original variable, let denote $\bar{\beta}_i$ and $S^2_{\hat{\beta}_i}$ as the sample mean and variance of the coefficients for its knockoff counterparts calculated as follows*

$$\bar{\beta}_i = \frac{1}{k_{max}} \sum_{k=1}^{k_{max}} \hat{\beta}_{k,i}, \quad S^2_{\hat{\beta}_i} = \frac{1}{k_{max} - 1} \sum_{k=1}^{k_{max}} (\hat{\beta}_{k,i} - \bar{\beta}_i)^2 \ . \tag{12}$$

The null (H_0) and alternative (H_1) hypotheses for testing the significance of the i^{th} original variable relevant to the response variable y are given as follows.

$$H_0 : \hat{\beta}_i \sim N(\bar{\beta}_i, S^2_{\hat{\beta}_i}), \quad H_1 : \hat{\beta}_i \not\sim N(\bar{\beta}_i, S^2_{\hat{\beta}_i}) \ . \tag{13}$$

Note that only coefficients of the knockoff variables are considered for distribution modeling, while the coefficient of the original variable is used for testing.

We have transformed variable selection into a hypothesis testing framework, enabling a systematic and quantifiable evaluation of statistical relevance and supporting informed decisions on variable inclusion.

3.3.2 Estimation of *p*-values based on Coefficient Differences

To implement the above significance test, this step employs a Z-statistic approach [24, 25] to estimate the p-value of the original features. Given the vector of coefficients, $\hat{\boldsymbol{\beta}}$, as the input, the Z-statistic approach first calculates the normalized difference between coefficients for the i^{th} original variable as follows.

$$\hat{Z}_i := \frac{\sqrt{k_{max} - 1}(\hat{\beta}_i - \bar{\beta}_i)}{S_{\hat{\beta}_i}}, \tag{14}$$

where $\hat{\beta}_i$ denotes the estimated coefficient of the i^{th} original variable, $\bar{\beta}_i$ and $S_{\hat{\beta}_i}$ have been defined in *Definition* 1. Later, we compute the p-value of the i^{th} original variable relevant to the response variable $\mathbf{y}$ using the equation as follows.

$$\hat{P}_i := 2\Phi(-|\hat{Z}_i|), \tag{15}$$

where Φ denotes the cumulative distribution function (CDF) of the standard normal distribution. A small p-value suggests a significant, non-random link between the variable and $\mathbf{y}$, guiding effective variable selection in **Knoop**.

3.3.3 Algorithm Analysis

We formulate our main analytical result as the control of False Discovery Rate (FDR) .

Proposition 2 (FDR Control of Anomaly-based Significance Test)**.** *Let V be the number of false positives (i.e., original variables falsely selected as significant) and R*

be the total number of original variables selected as significant by the anomaly-based significance test at a given threshold. The FDR of the test is defined as:

$$\text{FDR} := \mathbb{E}\left[\frac{V}{R} \mid R > 0\right] \cdot \mathbb{P}(R > 0)$$

Under assumptions A3 and A4, the anomaly-based significance test controls the FDR at level α if the p-values of the original variables are estimated using the Z-statistic approach and the Benjamini-Hochberg (BH) procedure [19] is applied to select significant variables based on the estimated p-values at a target FDR level α.

Proof. Under assumptions A3 and A4, we here prove that the test controls the FDR at level α when the p-values of the original variables are estimated using the Z-statistic approach and the Benjamini-Hochberg (BH) procedure is applied to select significant variables based on the estimated p-values at a target FDR level α.

Let p be the total number of original variables, and p_0 be the number of original variables for which the null hypothesis H_0 is true (i.e., variables not significantly related to the response). Let $P_1, P_2, \ldots, P_p$ be the true p-values of the original variables, and $\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_p$ be the estimated p-values using the Z-statistic approach.

Step 1: Under assumption A3, for any original variable i for which H_0 is true, the Z-statistic $\hat{Z}_i$ follows a standard normal distribution:

$$\hat{Z}_i \sim N(0,1) \text{ under } H_0 \ .$$

Step 2: For any original variable i for which H_0 is true, the estimated p-value $\hat{P}_i$ using the Z-statistic approach is uniformly distributed on $[0,1]$:

$$\hat{P}_i \sim U(0,1) \text{ under } H_0 \ .$$

This follows from the fact that $\hat{P}_i = 2\Phi(-|\hat{Z}_i|)$ and $\hat{Z}_i \sim N(0,1)$ under H_0. Furthermore, under assumption A4, the estimated p-values $\hat{P}_i$ for variables with true H_0 are independent of each other, as the coefficients of the knockoff counterparts for each original variable are assumed to be independent.

Step 3: The BH procedure, when applied to the estimated p-values $\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_p$ at a target FDR level α, controls the FDR at level α under the conditions established in steps 1 and 2. This is because the BH procedure controls the FDR at level α when the p-values corresponding to the true null hypotheses are uniformly distributed on $[0,1]$ and independent of each other [19].

Therefore, under assumptions A3 and A4, the anomaly-based significance test, which estimates p-values using the Z-statistic approach and applies the BH procedure to select significant variables, controls the FDR at level α. □

Remark on Proposition 2. The FDR control property ensures that, among the original variables selected as significant by the anomaly-based test, the expected proportion of false positives is at most α. This provides a guarantee for the effectiveness of the tests in identifying truly significant variables while controlling the false positive rate.

4 Experiments

This section uses both controlled simulations and applied real-world scenarios to evaluate the capabilities of **Knoop**, assessing the effectiveness in selecting variables for enhanced prediction.

Algorithm 4 Data Synthesis by Simulation

Input: Sample size n, parameter count p, real parameter count p_{real}, variable covariance $\Sigma \in \mathbb{R}^{p \times p}$, error variance σ^2.

Output: Generated data matrix X, training labels $\mathbf{y}$, coefficient vector β.

1: $X \leftarrow n \times p$ matrix with i.i.d. rows from $\mathcal{N}(\vec{0}, \Sigma)$
2: $\beta_1, \ldots, \beta_{p_{\text{real}}} \leftarrow$ entries $\overset{\text{i.i.d.}}{\sim} U(0,1)$
3: $\boldsymbol{\beta} \leftarrow \text{randomPermute}([\beta_1, \ldots, \beta_{p_{\text{real}}}, 0, \ldots, 0]^\top)$ ▷ Randomly permute coefficients
4: $\boldsymbol{\varepsilon} \leftarrow n$-dimensional vector with i.i.d. entries from $\mathcal{N}(0, \sigma^2)$
5: $\mathbf{y} \leftarrow X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$
6: **return** variable matrix X, response vector $\mathbf{y}$, coefficients $\boldsymbol{\beta}$.

4.1 Simulation with Synthesized Datasets

We setup two categories of experiments with the synthesized datasets based on simulations (in Algorithm 4) as follows.

- **Low-dimensional experiments:** These experiments investigate various parameter settings for low-dimensional data generated from simulations. The data matrix X is drawn from a multivariate normal distribution with covariance matrix $\Sigma_{i,j} = \rho^{|i-j|}$, where $\rho = 0.25$, and the error variance is set to $\sigma^2 = 1$. The experiments vary the number of variables $p \in \{80, 100, 150, 180\}$, non-zero components $p_{\text{real}} \in \{10, 20, 30, 40, 50\}$, and sample size $n \in \{85, 100, 120\}$. Each experimental setting is repeated at least 20 times to ensure robust results.
- **High-dimensional experiment:** The experiment investigates the performance of the proposed method on a dataset with $p = 1000$ variables, of which $p_{\text{real}} = 30$ are non-zero components in the ground-truth model. The training sample size is set to $n \in \{100, 1000, 3000\}$. Each row of the variable matrix X is generated from a multivariate normal distribution $\mathcal{N}(\vec{0}, \Sigma)$, where the covariance matrix is defined as $\Sigma_{i,j} = \rho^{|i-j|}$ with $\rho = 0.1$. The error variance is set to $\sigma^2 = 0.25$. The experiment is repeated at least 20 times to ensure robust results.

Significant variables in ground truth: Based on the simulation settings in Algorithm 4, the variables corresponding to $\beta_1,\ \beta_2, \ldots,\ \beta_{P_{real}}$ are *significant variables in ground truth*, as the response variable $\mathbf{y}$ is generated as the noisy linear combination of these variables with non-zero coefficients (lines 2–5 in Algorithm 4).

In our experiments, we employ various regression and variable selection methods, including Ridge, Lasso, ElasticNet, Model-X and Multiple Knockoff filters [10, 17], for evaluation. For simplicity, as the generated data are already normalized, we directly

compare the regression coefficients of Lasso, ElasticNet, and Ridge as the importance of variables. All these implementations are derived from `scikit-learn`, `knockpy` and `multiknockoffs`, while hyper-parameters are tuned best through cross-validation. Note that **Knoop** utilizes $\ell_{max} = 4$ layers of hierarchical knockoffs (consisting of a total of 15 sets of knockoff matrices), while Multiple Knockoff employs 25 knockoff matrices according to the package default settings. To evaluate the effectiveness for variable selection, we use Area Under Curve (AUC) of the Receiver Operating Characteristic (ROC) curve to report our experiment results. As shown in Algorithm 3, the raw output of **Knoop** consists of a ranking list of variables ordered by their p-values. The AUC measures the true positive rate (TPR) and false positive rate (FPR) as variables are sequentially selected from the ranking list.

Table 1: Results (AUC) for Simulation-based Evaluation (values at the **first places** and the second places). Settings 1: $p = 80$, $p_{real} = 10$, $n = 100$, 2: $p = 100$, $p_{real} = 10$, $n = 100$, 3: $p = 150$, $p_{real} = 10$, $n = 100$, 4: $p = 180$, $p_{real} = 10$, $n = 100$, 5: $p = 100$, $p_{real} = 20$, $n = 100$, 6: $p = 100$, $p_{real} = 30$, $n = 100$, 7: $p = 100$, $p_{real} = 40$, $n = 100$, 8: $p = 100$, $p_{real} = 50$, $n = 100$, 9: $p = 100$, $p_{real} = 10$, $n = 85$, 10: $p = 100$, $p_{real} = 10$, $n = 120$, 11: $p = 1000$, $p_{real} = 30$, $n = 3000$, 12: $p = 1000$, $p_{real} = 30$, $n = 100$ and 13: $p = 1000$, $p_{real} = 30$, $n = 1000$.

Settings	**Knoop**	**Model-X K.**	**Multiple K.**	**Ridge**	**Lasso**	**ElasticNet**
The Low-Dimensional Experiments						
1	$\mathbf{0.788 \pm 0.068}$	0.783 ± 0.075	0.733 ± 0.062	0.700 ± 0.106	0.520 ± 0.000	0.616 ± 0.076
2	$\mathbf{0.784 \pm 0.063}$	0.776 ± 0.071	0.733 ± 0.063	0.603 ± 0.090	0.282 ± 0.000	0.388 ± 0.083
3	$\mathbf{0.796 \pm 0.071}$	0.777 ± 0.082	0.732 ± 0.065	0.740 ± 0.076	0.663 ± 0.000	0.708 ± 0.041
4	$\mathbf{0.792 \pm 0.065}$	0.778 ± 0.067	0.741 ± 0.059	0.764 ± 0.059	0.522 ± 0.003	0.603 ± 0.051
5	0.696 ± 0.060	0.688 ± 0.057	$\mathbf{0.734 \pm 0.060}$	0.552 ± 0.075	0.456 ± 0.000	0.477 ± 0.025
6	0.663 ± 0.045	0.650 ± 0.059	$\mathbf{0.738 \pm 0.062}$	0.534 ± 0.055	0.408 ± 0.003	0.431 ± 0.027
7	0.646 ± 0.051	0.638 ± 0.050	$\mathbf{0.748 \pm 0.060}$	0.526 ± 0.059	0.595 ± 0.000	0.607 ± 0.013
8	0.621 ± 0.051	0.617 ± 0.052	$\mathbf{0.744 \pm 0.611}$	0.529 ± 0.050	0.530 ± 0.000	0.533 ± 0.012
9	$\mathbf{0.765 \pm 0.079}$	0.752 ± 0.073	0.745 ± 0.056	0.648 ± 0.092	0.282 ± 0.000	0.398 ± 0.085
10	$\mathbf{0.803 \pm 0.067}$	0.791 ± 0.063	0.742 ± 0.059	0.700 ± 0.091	0.282 ± 0.000	0.384 ± 0.095
The High-Dimensional Experiment						
11	$\mathbf{0.985 \pm 0.011}$	0.981 ± 0.013	0.737 ± 0.056	0.969 ± 0.016	0.366 ± 0.000	0.366 ± 0.000
12	0.677 ± 0.003	0.675 ± 0.003	$\mathbf{0.746 \pm 0.047}$	0.719 ± 0.002	0.507 ± 0.007	0.532 ± 0.007
13	0.894 ± 0.001	$\mathbf{0.902 \pm 0.001}$	0.736 ± 0.059	0.721 ± 0.004	0.421 ± 0.000	0.421 ± 0.000

4.1.1 Experiment Results

The experiment results for variable selection, as presented in Table 1, showcase the performance of different methods in both low-dimensional and high-dimensional settings. Across all 13 experimental settings, **Knoop** achieves the highest AUC values in 7 of them and takes the second places in 5 of them, indicating its superior ability to prioritize significant variables. Especially, for the low-dimensional experiments (settings 1-10), **Knoop** always achieves the highest or the second highest AUC values. Similarly, in the high-dimensional experiments (settings 11-13), **Knoop** maintains its

superior performance, particularly when the sample size is sufficient (settings 11 and 13). These results highlight the ability of **Knoop** to accurately identify and rank significant variables, across different numbers of variables (p), non-zero components (p_{real}), and sample sizes (n).

4.1.2 Discussions

Knoop and Knockoff methods generally outperform Lasso or ElasticNet in variable selection, achieving higher AUC, due to their specific design to control FDRs while maintaining power via so-called "selective inference" [26]. Utilizing coefficients from Lasso and ElasticNet for variable selection can even result in an AUC below 0.5, indicative of performance inferior to random guessing. Methods like Lasso or ElasticNet can still fit and predict data well, even when their non-zero coefficients are not assigned to the true variables, due to the inconsistency in coefficient estimates [27]. Specifically, in high-dimensional scenarios where an extremely sparse model is fitted, almost all the coefficients are estimated as 0. Consequently, all variables with zero coefficients may be considered to be equally irrelevant, making it impracticable to distinguish or select meaningful variables based solely on their coefficients.

Table 2: Overview of Datasets Used for Model Evaluation

Dataset	**#Variables**	**#Train/Test**	**Prediction Task**
Alon Dataset [28]	2000	37/25	Classification (genetics)
Communities and Crime [29]	100	21/9	Regression (sociology)
Superconductivity [30]	81	360/40	Regression (chemistry)
Appliances Energy Prediction [20]	27	225/25	Regression (environment)

4.2 Evaluation with Realistic Datasets

Table 2 provides the profile of every dataset for evaluation, including the number of variables, the number of samples, and the context of tasks. To carry out the experiments, we use following two categories of baseline algorithms for comparisons.

- Statistical methods such as model-X Knockoff, multiple Knockoff, Ridge regression, Lasso, and ElasticNet are utilized to select variables by their statistical significance. For every comparison here, **Knoop** and these methods are set to selected a fixed-length of variables in every experiment (please refer to Section 3.1).
- Global optimization-based feature selectors, including Harris Hawk Optimization (HHO) [31], Jaya Algorithm [32], Sine Cosine Algorithm (SCA) [33], Salp Swarm Algorithm (SSA) [34], and Whale Optimization Algorithm (WOA) [35] are used to search the subset of features. In the experiment, the number of variables to be selected by **Knoop** is determined via cross-validation for fair comparison, as all global optimization-based methods leverage the cross-validated performance measure as the search objective for feature selection (please refer to Section 3.1).

Note that **Knoop** employs a hierarchical knockoff structure spanning $\ell_{max} = 3$ layers (totaling 7 sets of knockoff matrices). For classification tasks, we use misclassification rate for evaluation and comparison. For regression tasks, we report mean squared error (MSE) as the measurements of performance.

Table 3: Result comparisons with statistical methods for fixed-length selection (values at the **first places** and the second places)

#Selected Variables	**Knoop**	**Model-X K.**	**Multiple K.**	**Ridge**	**Lasso**	**ElasticNet**	**Lars**
Misclassification Rate on Alon Dataset							
1	**0.400**	0.600	**0.400**	**0.400**	**0.400**	**0.400**	**0.400**
2	**0.120**	0.320	0.160	0.400	0.440	0.400	0.360
4	**0.160**	0.240	0.320	0.280	0.280	0.400	0.360
8	0.240	0.280	0.360	0.240	**0.160**	0.240	0.360
MSE on Communities and Crime Dataset ($\times 10^{-7}$)							
1	**3.147**	24.200	24.200	2084	24.200	24.200	5869
2	**5.447**	24.440	272.070	2187	24.440	24.440	5120
4	**63.204**	17630	8745	2522	17630	17630	196100
8	**401.5**	88530	8799	9213	88530	88530	182800
MSE on Superconductivity Dataset ($\times 10^{2}$)							
1	**5.003**	9.023	6.267	9.730	9.847	9.847	7.034
4	**2.714**	9.874	4.358	11.125	6.034	6.462	5.049
8	3.047	8.826	2.436	4.850	3.095	**2.348**	3.064
MSE on Appliances Energy Prediction Dataset ($\times 10^{3}$)							
1	**3.861**	5.527	**3.861**	5.527	5.527	5.321	5.624
2	**3.855**	5.357	4.295	5.079	5.357	5.214	5.160
4	**3.720**	4.922	4.444	4.860	5.066	5.834	5.059
8	**3.895**	6.581	4.504	5.109	6.581	4.243	4.793

4.2.1 Experiment Results

In Table 3, we present the comparison results between **Knoop** and statistical variable selection methods under the settings of fixed-length selection (1–10 selected variables). Due to the page length, we only present 1, 2, 4, and 8 here. Specifically, we assess the misclassification rates of chosen variables through logistic regression model, alongside analyzing the MSE of these variables when employing linear regression for prediction. The results demonstrate that **Knoop** consistently outperforms the other methods in terms of misclassification rate and MSE when selecting a small number of variables (1, 2, or 4). This superior performance is particularly evident in the Communities and Crime dataset, where **Knoop** achieves significantly lower MSE values compared to the other methods. However, as the number of selected variables increases to 8, the performance of **Knoop** becomes comparable to or slightly worse than some of the other methods, such as Lasso and ElasticNet, in certain datasets like the Alon and

Table 4: Comparisons with global optimization-based methods for varying-length selection (values at the **first places** and the second places).

	Knoop	**HHO**	**Jaya**	**SCA**	**SSA**	**WOA**
Misclassification Rate on Alon Dataset						
Log. Reg.	**0.160**	0.400	0.200	0.320	0.240	0.240
Rand. Forest	**0.160**	0.320	0.200	0.280	**0.160**	0.400
GBDT	0.320	0.480	0.240	0.280	**0.160**	0.320
SVM-Linear	**0.200**	0.440	**0.200**	**0.200**	0.280	0.360
SVM-Radial	**0.200**	0.240	0.320	0.400	0.320	0.560
SGDClassifer	0.240	0.240	0.200	**0.160**	0.200	0.200
MSE on Communities and Crime Dataset ($\times 10^{-3}$)						
Lin. Reg.	$\mathbf{3.147 \times 10^{-4}}$	6.473×10^{-2}	9.513	6.342×10^{-3}	8.156	0.130
Rand. Forest	$\mathbf{1.733 \times 10^{-30}}$	0.811	4.887	0.943	0.182	3.591
GBDT	$\mathbf{5.420 \times 10^{-13}}$	4.004×10^{-12}	0.769	0.256	0.120	13.98
SVM-Linear	1.947	1.486	2.263	**0.574**	1.194	4.618
SVM-Radial	3.700	1.169	3.327	1.683	3.170	**0.740**
Neural Net (2)	4.808×10^{-2}	5.676×10^{-3}	9.548×10^{-2}	$\mathbf{5.656 \times 10^{-3}}$	5.893×10^{-3}	6.363×10^{-3}
Neural Net (3)	5.676×10^{-3}	5.357×10^{-3}	2.601×10^{-2}	5.739×10^{-3}	$\mathbf{3.595 \times 10^{-3}}$	5.698×10^{-3}
MSE on Superconductivity Dataset ($\times 10^{1}$)						
Lin. Reg.	**13.594**	41.925	18.036	67.446	58.543	53.824
Rand. Forest	13.248	7.429	11.125	7.418	**7.070**	8.032
GBDT	8.452	8.596	**7.712**	12.410	9.211	9.030
SVM-Linear	**31.634**	65.979	468.9	33.935	71.169	71.979
SVM-Radial	**36.210**	62.506	70.990	73.750	74.102	68.142
Neural Net (2)	62.685	78.623	215.5	**57.320**	61.703	60.868
Neural Net (3)	**51.826**	61.643	61.142	54.958	81.451	59.789
MSE on Appliances Energy Prediction Dataset ($\times 10^{3}$)						
Lin. Reg.	10.303	2.331	**2.310**	4.904	4.242	4.590
Rand. Forest	11.103	8.608	**3.939**	6.577	4.904	5.392
SVM-Linear	4.480	4.909	4.855	5.091	**4.479**	4.957
SVM-Radial	**4.600**	5.162	4.966	5.166	5.160	4.910
Neural Net (2)	5.202	5.340	**5.191**	5.291	5.224	5.216
Neural Net (3)	5.231	**5.216**	5.258	5.347	5.291	8.644

Superconductivity datasets. Note that the misclassification rate on Alon dataset could be even higher than 0.5, as the samples are extremely class-imbalanced.

Table 4 compares the performance of **Knoop** against several global optimization-based feature selectors for the four datasets, under varying-length selection settings. The results demonstrate that **Knoop** not only achieves the highest accuracy (measured by misclassification rate or MSE) when selecting features for a specific machine learning algorithm on the majority of datasets but also almost delivers the highest accuracy across all machine learning models for each dataset. We are encouraged to see **Knoop** outperforms the global optimization-based approaches for feature selection, as these methods leveraged cross-validated accuracy as objectives to directly search the subset of features, for predictive modeling.

Table 5: Comparison of runtime (in seconds) between methods across different datasets.

Method\Dataset	Alon	Communities.	Superconductivity	Appliances.
Knockoff	56.344	0.065	0.048	0.015
Knoop	2266.665	1.387	1.143	0.225
Multiple Knockoff	19360.112	2.734	3.436	0.839

4.2.2 Discussions on Performance Comparisons

Comparing to sparse linear models and global optimization-based feature selectors, the core advantage of **Knoop** (as well as the Knockoff baselines) is its capability in testing the conditional independence between variables and the response for prediction purposes [7]. Apparently, filtering the variables that are conditional independent with the responses is more effective than just estimating the sparsest set of coefficients, resulting in more generalizable models and preventing models learning from noise as signal. Furthermore, such test helps in distinguishing causation from correlation. This distinction is crucial in scenarios where understanding causal relationships is as important as achieving high prediction accuracy [11]. Finally, **Knoop** outperforms existing Knockoff methods by generating multiple sets of knockoff variables and integrating them into a Ridgeless regression to enhance capacity. It evaluates the significance of each original variable by comparing its coefficient against the distribution of its knockoff counterparts, utilizing a larger reference group of null variables, offering a more precise assessment of impact of original variables in prediction.

Of-course, our experiments are with some limitations. For example, we have not compare **Knoop** with some computation-intensive neural feature selectors such as sparse autoencoder [5]. Moreover, though **Knoop** provides an estimate of p-value, we have not evaluate the feature selection procedure by thresholding p-values (such as ≤ 0.05). Actually, we have done some trials by setting the threshold to ≤ 0.05 for the real datasets. **Knoop** can deliver a test set MSE of 4.015×10^{-5} with the Communities and Crime dataset, and 3.047×10^{2} with the Superconductivity dataset, even without cross-validation to determine the number of selected variables.

4.2.3 Discussions on Complexity and Time Cost

Here, we discuss the computational complexity and cost of our methods. Specifically, we analyze comparisons between **Knoop**, vanilla Knockoff, and multiple Knockoff methods. All three approaches require the generation of a data matrix of knockoff variables for inference. Given a p-dimensional dataset with n samples, the vanilla Knockoff method needs to generate a $p \times n$ data matrix. In addition, the multiple Knockoff method involves multiple rounds of Knockoff procedures for statistical inference. Let's denote the number of rounds for multiple Knockoff as R_{ound} and it results in the generation of R_{ound} knockoff data matrices and each of which is sized $p \times n$. In contrast, **Knoop** generates ℓ-layered knockoff matrices, necessitating the creation of a knockoff data matrix sized $((2^{\ell} - 1) \cdot p) \times n$. It is evident that the size of the data matrix for knockoff variables increases exponentially with the number of layers ℓ in

Knoop, while it increases linearly with the number of rounds in the multiple Knockoff method. However, in our experiments, we found that achieving equivalent or better performance usually requires a quite small ℓ, making $(2^\ell - 1)$ smaller than R_{ound}. For instance, in all experiments presented in Table 3, we set $\ell = 3$ (i.e., $2^\ell - 1 = 7$) while the number of multiple knockoff rounds is set to $R_{\text{ound}} = 25$.

Furthermore, considering time consumption, the cost to generate a $p \times n$ knockoff data matrix varies and can be expressed as $O(p^3 + p^2 \cdot n)$ when using Metropolized Knockoff [21]. In this way, the complexity of **Knoop** would be $O((2^\ell - 1)^3 \cdot p^3 + (2^\ell - 1)^2 \cdot p^2 \cdot n)$ with the Metropolized Knockoff[1]. Table 5 presents the time consumption of vanilla Knockoff, multiple Knockoff, and **Knoop**. Note that the vanilla Knockoff and **Knoop** were implemented with Knockpy[2] under the same settings, while multiple Knockoff was directly imported from the open-source solution[3]. The results indicate that although the complexity of **Knoop** may increase exponentially with the number of layer ℓ, it still takes less time to execute compared to multiple Knockoff. More importantly, the overall time consumption is relatively low when the number of dimensions is low (e.g., in datasets for communities, superconductivity, and appliances). Even for the Alon dataset, which has 2000 dimensions, the time consumption remains manageable, particularly when compared to multiple Knockoff.

5 Discussion and Conclusions

This work introduces **Knoop** to enhance variable selection, which addresses the model capacity issue in regression models used in the common Model-X Knockoff filter. The proposed approach first generates multiple sets of knockoff variables through *recursive generation*, and then integrates every original variable and its multiple knockoff counterparts into a *Ridgeless regression*. **Knoop** refines coefficient estimates with better capacity, and employs an *anomaly-based significance test* for robust variable selection. **Knoop** can either pickup a predefined number of variables or optimizes the number of selected variables by cross-validation. Extensive experiments demonstrate superior performance compared to existing methods in relevant variable discovery against ground truth variables controlled by simulations and supervised feature selection for classification/regression tasks.

Declaration

- Funding - Not applicable
- Conflicts of interest/Competing interests - Not applicable
- Ethics approval - No data have been fabricated or manipulated to support your conclusions. No data, text, or theories by others are presented as if they were our

[1] In contrast, Gimenez *et al.* [13] didn't disclose the complexity of multiple knockoff in their work. However, it incrementally utilizes the Sequential Conditional Independence Pairs (SCIP)–the time consuming step in Model-X Knockoff–to generate multiple sets of Knockoff variables and ensure the exchangeability in the joint distribution (i.e., between every original variable and its knockoff counterpart and between any two knockoff variables corresponding to the same original variable). Thus its computation cost should be high when R_{ound} is large.

[2] https://github.com/amspector100/knockpy

[3] https://github.com/cKarypidis/multiknockoffs

own. Data we used, the data processing and inference phases do not contain any user personal information. This work does not have the potential to be used for policing or the military.

- Consent to participate - Not applicable
- Consent for publication - Not applicable
- Availability of data and material - Experiments are based on publicly available open-source datasets.
- Code availability - The codes are available at https://github.com/RubyZhang166/Knoop-Knockoff-Enhancement-with-Overparametrization-for-Feature-Selection.
- Authors' contributions - H.X contributed the original idea. X.Z conducted experiments. X.Z and H.X wrote the manuscript. Y.C involved in discussion and wrote part of the manuscript. X.Z and H.X share equal technical contribution.

References

[1] He, Z., Liu, L., Belloy, M.E., Le Guen, Y., Sossin, A., Liu, X., Qi, X., Ma, S., Gyawali, P.K., Wyss-Coray, T., *et al.*: Ghostknockoff inference empowers identification of putative causal variants in genome-wide association studies. Nature Communications **13**(1), 7209 (2022)

[2] Tibshirani, R.: Regression shrinkage and selection via the lasso. JRSS B **58**(1), 267–288 (1996)

[3] Frandi, E., Ñanculef, R., Lodi, S., Sartori, C., Suykens, J.A.: Fast and scalable lasso via stochastic frank–wolfe methods with a convergence guarantee. Mach. Learn. **104**, 195–221 (2016)

[4] Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. JRSS B **67**(2), 301–320 (2005)

[5] Atashgahi, Z., Sokar, G., Lee, T., Mocanu, E., Mocanu, D.C., Veldhuis, R., Pechenizkiy, M.: Quick and robust feature selection: the strength of energy-efficient sparse training for autoencoders. Mach. Learn., 1–38 (2022)

[6] Ge, Y., Sealfon, S.C., Speed, T.P.: Multiple testing and its applications to microarrays. Statistical Methods in Medical Research **18**, 543–563 (2009)

[7] Watson, D.S., Wright, M.N.: Testing conditional independence in supervised learning algorithms. Mach. Learn. **110**(8), 2107–2129 (2021)

[8] CANDES, E., TAO, T.: The dantzig selector: Statistical estimation when p is much larger than n. Ann. Stat. **35**(6), 2313–2351 (2007)

[9] Barber, R.F., Candès, E.J.: Controlling the false discovery rate via knockoffs. Ann. Stat. **43**(5), 2055–2085 (2015)

[10] Candes, E., Fan, Y., Janson, L., Lv, J.: Panning for gold:'model-x'knockoffs for high dimensional controlled variable selection. JRSS B **80**(3), 551–577 (2018)

[11] Yu, K., Guo, X., Liu, L., Li, J., Wang, H., Ling, Z., Wu, X.: Causality-based feature selection: Methods and evaluations. ACM Computing Surveys (CSUR) **53**(5), 1–36 (2020)

[12] Hoerl, A.E., Kennard, R.W.: Ridge regression: Biased estimation for nonorthogonal problems. Technometrics **12**(1), 55–67 (1970)

[13] Gimenez, J.R., Zou, J.: Improving the stability of the knockoff procedure: Multiple simultaneous knockoffs and entropy maximization. In: AISTATS, pp. 2184–2192 (2019). PMLR
[14] Katsevich, E., Sabatti, C.: Multilayer knockoff filter: Controlled variable selection at multiple resolutions. The annals of applied statistics **13**(1), 1 (2019)
[15] LIANG, T., RAKHLIN, A.: Just interpolate: Kernel "ridgeless" regression can generalize. Ann. Stat. **48**(3), 1329–1347 (2020)
[16] Tsigler, A., Bartlett, P.L.: Benign overfitting in ridge regression. JMLR **24**(1) (2024)
[17] Nguyen, T.-B., Chevalier, J.-A., Thirion, B., Arlot, S.: Aggregation of multiple knockoffs. In: International Conference on Machine Learning, pp. 7283–7293 (2020). PMLR
[18] Hastie, T., Montanari, A., Rosset, S., Tibshirani, R.J.: Surprises in high-dimensional ridgeless least squares interpolation. Ann. Stat. **50**(2), 949 (2022)
[19] Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. JRSS B **57**(1), 289–300 (1995)
[20] Candanedo, L.M., Feldheim, V., Deramaix, D.: Data driven prediction models of energy use of appliances in a low-energy house. Energy and Buildings **140**, 81–97 (2017)
[21] Bates, S., Candès, E., Janson, L., Wang, W.: Metropolized knockoff sampling. JASA **116**(535), 1413–1427 (2021)
[22] Romano, Y., Sesia, M., Candès, E.: Deep knockoffs. JASA **115**(532), 1861–1872 (2020)
[23] Richards, D., Mourtada, J., Rosasco, L.: Asymptotics of ridge (less) regression under general source condition. In: AISTATS, pp. 3889–3897 (2021). PMLR
[24] Clogg, C.C., Petkova, E., Haritou, A.: Statistical methods for comparing regression coefficients between models. American Journal of Sociology, 1261–1293 (1995)
[25] Paternoster, R., Brame, R., Mazerolle, P., Piquero, A.: Using the correct statistical test for the equality of regression coefficients. Criminology **36**(4), 859–866 (1998)
[26] Barber, R.F., Candès, E.J.: A knockoff filter for high-dimensional selective inference. Ann. Stat. **47**(5), 2504–2537 (2019)
[27] Lee, J.H., Shi, Z., Gao, Z.: On lasso for predictive regression. Journal of Econometrics **229**(2), 322–349 (2022)
[28] Alon, U., Barkai, N., Notterman, D.A., Gish, K., *et al.*: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. PNAS **96**(12), 6745–6750 (1999)
[29] Redmond, M., Baveja, A.: A data-driven software tool for enabling cooperative information sharing among police departments. European Journal of Operational Research **141**(3), 660–678 (2002)
[30] Hamidieh, K.: A data-driven statistical model for predicting the critical temperature of a superconductor. Computational Materials Science **154**, 346–354 (2018)

[31] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H.: Harris hawks optimization: Algorithm and applications. Future generation computer systems **97**, 849–872 (2019)
[32] Houssein, E.H., Gad, A.G., Wazery, Y.M.: Jaya algorithm and applications: A comprehensive review. Metaheuristics and Optimization in Computer and Electrical Engineering, 3–24 (2021)
[33] Mirjalili, S.: Sca: a sine cosine algorithm for solving optimization problems. Knowledge-based systems **96**, 120–133 (2016)
[34] Abualigah, L., Shehab, M., Alshinwan, M., Alabool, H.: Salp swarm algorithm: a comprehensive survey. Neural Computing and Applications **32**(15), 11195–11215 (2020)
[35] Mirjalili, S., Lewis, A.: The whale optimization algorithm. Advances in engineering software **95**, 51–67 (2016)