Error-controlled feature selection for ultrahigh-dimensional and highly correlated feature space using deep learning

Arkaprabha Ganguli¹ | Tapabrata Maiti¹ | David Todem²

¹Department of Statistics & Probability, Michigan State University, East Lansing, Michigan, USA
²Department of Epidemiology & Biostatistics, Michigan State University, East Lansing, Michigan, USA

Correspondence
Arkaprabha Ganguli, Department of Statistics & Probability, Michigan State University, East Lansing, MI, USA. Email: aganguli@anl.gov

Present address
Arkaprabha Ganguli, Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, Illinois, USA

Funding information
National Science Foundation, Grant/Award Number: DMS1916339

Abstract
Deep learning has been at the center of analytics in recent years due to its impressive empirical success in analyzing complex data objects. Despite this success, most existing tools behave like black-box machines, thus the increasing interest in interpretable, reliable, and robust deep learning models applicable to a broad class of applications. Feature-selected deep learning has emerged as a promising tool in this realm. However, the recent developments do not accommodate ultrahigh-dimensional and highly correlated features or high noise levels. In this article, we propose a novel screening and cleaning method with the aid of deep learning for a data-adaptive multi-resolutional discovery of highly correlated predictors with a controlled error rate. Extensive empirical evaluations over a wide range of simulated scenarios and several real datasets demonstrate the effectiveness of the proposed method in achieving high power while keeping the false discovery rate at a minimum.

KEYWORDS
deep learning, false discovery rate, high dimensional feature selection, multicollinearity, sparsity-induced neural network

1 | INTRODUCTION

In modern applications (e.g., genetics and imaging studies), the investigator is often interested in uncovering the true pattern between a quantitative response and a large number of features. The key working assumption, oftentimes, is that there is an underlying sparsity pattern buried in the high-dimensional data structure. Detecting the essential features plays a crucial role in facilitating scientific investigations by offering improved interpretability and explainability, coupled with reduced computational cost and less memory usage. Under the linear model (LM) framework, this problem has been extensively studied over the past few decades, stemming from the popular algorithms such as Lasso, Elastic net, SCAD, and MCP. A detailed review of this literature can be found elsewhere (e.g., [18]) and thus is omitted here. However, regardless of their ubiquitous applications, the LM has limited applications, especially when the underlying mechanism is highly nonlinear, with potential interaction effects. This is particularly critical given that the true underlying relationship is unknown to the analysts in real applications. The artificial neural network (ANN) models that relax the linearity assumption are well known for efficiently
approximating complicated functions. For example, from an information-theoretic viewpoint, Elbrächter et al. [16] established that deep neural networks (DNN) provide an optimal approximation of a nonlinear function, covering a wide range of functional classes used in signal processing. This property has promoted the use of deep learning (DL) models for feature selection, an approach that has generated much research interest over the past few years. However, the black-box nature and lack of explainability in DL models pose a major drawback, raising concerns about their applicability in real-world decision-making, as highlighted by Rudin [39]. Hence, it is crucial to exercise caution when employing DL models in practical applications.

1.1 Error-controlled feature selection

Employing only the relevant predictors to construct a predictive model is the right step toward explainable machine learning. Moreover, recent studies have shown that the feature importance in DL-based algorithms can demonstrate substantial variability in the presence of small input perturbations or a lower signal-to-noise ratio Ghorbani et al. [20]. To address this challenge, our approach aims to achieve reproducible nonlinear error-controlled variable selection using DL models. In this regard, we employ the false discovery rate (FDR), originally proposed by Benjamini & Hochberg [3]. FDR is well suited for large-scale multiple-testing problems, offering a less conservative and more powerful alternative compared with traditional methods such as the family-wise error rate (FWER). To formally define the FDR, we consider the random variable, FDP, representing the false discovery proportion: $FDP = \frac{e_0}{\max(N_0,1)}$, where $e_0$ = number of falsely selected variables and $N_0$ = number of total discoveries. Then, FDR is defined as $FDR = E(FDP)$. The estimation of this expectation, considering the unknown data-generating distribution, presents a distinct challenge in the context of model-free variable selection. To tackle this issue, various approaches have been proposed in the literature. For instance, multiple testing literature suggests using a $p$-value-based approach as a criterion for feature importance; refer to Tansey et al. [45]; Xia et al. [50]; Li & Barber [29]; Lei & Fithian [27] for an overview. However, generating interpretable $p$-values remains an unresolved research issue in the context of DL models. To circumvent this limitation, the knockoff framework has been proposed by Candès et al. [8]. Essentially, this is a model-free variable selection algorithm with provable FDR control, assuming one has prior knowledge of the predictors' distribution. Lu et al. [52] further proposed the DeepPIN$K$ algorithm by integrating the knockoff framework with the DL architecture for improved explainability of the DL models. Nevertheless, in real-world applications, it is often unrealistic to assume full knowledge of the distribution of the high-dimensional feature vector, which is a fundamental requirement for the knockoff approach. Consequently, the predictor's distribution needs to be estimated in a real-world application to generate the knockoff variables, which adds another layer of uncertainty to the analysis. Recently Barber et al. [1] showed that the knock-off framework can lead to inflation in false discoveries, which is proportional to the error incurred in estimating the predictor’s distribution. This problem is exacerbated by highly correlated features. Figure 1 provides an empirical illustration of Barber et al. [1], showing how the model-X knockoff approach of Candès et al. [8] fails to effectively control the FDR under high multicollinearity. Figure 1 also demonstrates that the problem lies in accurately estimating the distribution of the features, as knockoffs generated from the true feature distribution successfully control the FDR. However, relying on the true distribution is often unrealistic in practical settings where the feature distribution is typically unknown. Consequently, any knockoff generation method that imposes distributional assumptions on the features, such as [8, 40], may prove inefficient in a general setting. To address these challenges associated with model-based knockoff methods, recent advancements have introduced flexible knockoff generation algorithms based on deep learning (DL), such as [24, 33, 38]. However, they are trained in a typical big-$n$-small-$p$ setting, and their performance in ultrahigh-dimensional settings with limited sample size remains unclear. We next discuss in detail the multicollinearity issue in the context of multi-resolutional feature selection.

1.2 Addressing uncertainty in highly correlated feature space

In contemporary high-dimensional datasets often found in genetics and imaging studies, the presence of extreme multicollinearity is a significant challenge in addition to ultrahigh dimensionality. This phenomenon manifests as a complex intercorrelation among the predictors, often with pairwise sample correlations surpassing 0.99. In a simplistic linear model setting, Figure 2 shows how increased autocorrelation typically reduces the selection probability of the true nonnull features. Due to the near-indistinguishability of extremely correlated features, it is unrealistic to assert that a specific feature within a cluster is independently associated with the outcome in a regression framework. Therefore, it is more pragmatic
FIGURE 1 Examining the impact of multicollinearity on knockoffs—A demonstration through a simplistic simulation setting. We generate \( n = 400 \) independent and identically distributed samples \( (y, X) \in \mathbb{R} \times \mathbb{R}^{100} \), where the outcome \( y \) is generated by the linear model:

\[
y = 0.5(X_{20} + X_{40} + X_{60} + X_{80} + X_{100}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, 20),
\]

The features \( X \) follow a multivariate normal distribution \( N_{100}(0, \Sigma) \), where \( (\Sigma)_{ij} = \rho^{|i-j|} \) represents the autocorrelation strength. We vary the autocorrelation strength \( \rho \) from 0.1 to 0.95. Two approaches for Model-X knockoff generation are implemented on 500 Monte Carlo replication of the data: (1) Model-X estimated, using the method proposed in Candès et al. [8] to generate knockoffs based on the estimated feature distribution, and (2) Model-X True, generating knockoffs from the true distribution of \( X \). The FDR (false discovery rate) is depicted by vertical bars along the left y-axis, while the associated power is represented by horizontal lines on the right y-axis. The black horizontal line indicates the nominal FDR control level at \( q = 0.20 \).

to address the uncertainty by pursuing group-level variable selection, asserting that at least one variable within a densely correlated group is significant for the outcome. As such, by “true discovery,” one refers to the selection of clusters that can serve as good proxies for at least one element in the true index set of important features \( S_0 \). This approach has been discussed in the context of statistical genetics. For example, Brzyski et al. [5] eloquently argued that the discovery of a specific genomic region is considered more informative than the discovery of individual variants within that region. Although conceptually appealing, a complication of this approach is that the notion of FDR in its original formulation becomes nontrivial. For this reason, following Siegmund et al. [43], we adopt the cluster version of the FDR as the expected value of the “proportion of clusters that are falsely declared among all declared clusters.” We denote this as \( cFDR \) henceforth. There are other methods for handling extreme multicollinearity, which have for the most part appeared in the hierarchical testing literature. Few examples include CAVIAR [23], SUSIE [48], and KnockoffZoom [41]. While the knockoff-based procedures have the limitation of generating knockoffs from an unknown distribution with a very small sample size, other methods lack applicability in nonlinear–nonparametric setups as they typically depend on p-values.

1.3 Our contribution

To address the aforementioned complexities and unexplored gaps in variable selection, particularly when applying deep learning (DL) methods, we introduce the SciDNet-Screening & Cleaning Incorporated Deep Neural Network. This novel approach enables reproducible high-dimensional nonlinear-nonparametric feature selection in the presence of highly correlated predictors. The method consists of two steps: screening and cleaning. In the screening step, we perform dimension reduction by eliminating null features and constructing a set of multi-resolution clusters. These clusters collectively encompass all proxy variables necessary to capture the truly significant features with a high probability. In the cleaning step, we employ a carefully tuned DL model with an appropriate resampling scheme to estimate the FDR. This estimation serves as a key component for selecting
clusters of highly correlated predictors while controlling the FDR. To this end, FDR observed for “SciDNet” would implicitly refer the value of the cFDR discussed above. Our major contributions are summarized below:

- The proposed method, SciDNet, incorporates techniques from statistical machine learning and sparse modeling with the DL framework. It introduces a resampling-based FDR estimation scheme, allowing for the identification of relevant features while discarding irrelevant ones in an FDR-controlled setting. Moreover, SciDNet is specifically designed to handle the challenges posed by highly correlated features, which are known to be problematic for traditional feature selection methods.
- To the best of our knowledge, no other method in the literature addresses the multicollinearity issue in a high-dimensional setting through data-adaptive cluster formation, followed by nonlinear-nonparametric error-controlled feature selection integrated with DL. Extensive simulations and real data analyses demonstrate the validity and efficacy of our proposed method. Additionally, an ablation study dissects the contributions of individual steps in the overall performance of the feature selection method.
- Unlike existing state-of-the-art methods, our approach relies on minimal modeling assumptions and does not rely on p-values. As such, it provides a robust and better understanding of the sparse relationship between the outcome and the high-dimensional predictors. Through theoretical analysis, we establish the provable FDR-control guarantee of SciDNet in an asymptotic setting, further consolidating the empirical results.

In summary, our contributions offer a valuable tool to address the intricate task of selecting pertinent variables from highly correlated ultrahigh-dimensional datasets across diverse fields, including but not limited to biology, finance, and social sciences. The subsequent sections of this article are organized as follows: Section 2 provides an in-depth explanation of the proposed screening and cleaning method, while Section 3 delves into a theoretical discussion regarding the guaranteed control of FDR. This is followed by an extensive simulation study outlined in Section 4, as well as comprehensive analyses of two
real-world gene-expression datasets featured in Section 5. Finally, Section 6 serves as a conclusion, summarizing our findings and highlighting potential future research directions. Additional technical details and supplementary simulation results are presented in the appendix.

## 2 | THE ALGORITHM

### 2.1 | Notation and assumptions

Under the supervised learning framework, let $Y$ denote a continuous response variable, and $X = (X_1, \ldots, X_p)$ denote $p$ continuous covariates. Let $F_y(\cdot)$ denote the CDF of the response variable $Y$, and let $F_k(\cdot)$ denote the CDF of the predictor $X_k$. The dataset comprises $n$ independent and identically distributed (iid) samples, each represented as $(Y_i, X_i) = (Y_i, X_{i1}, X_{i2}, \ldots, X_{ip})$, where $i$ refers to the data point index. Notably, we focus on the ultrahigh-dimensional setting where the number of predictors, $p$, grows exponentially with $n$ as $p = O(\exp(n^\tau))$, with $\tau > 0$. We assume no specific functional relationship between the outcome $Y$ and the predictors $X$, but we impose a high-level assumption on the distribution of $X$. In particular, we assume that the predictors follow the nonparanormal distribution; i.e., there exist unknown differentiable functions $g(X) = \{g_j(X_j), j \in \{1, 2, \ldots, p\}\}$, such that $g(X) \sim N(\mu^{\#X}, \Sigma^{\#X})$. This nonparanormal distribution encompasses a wide range of parametric families of distributions and maintains the conditional dependency structure of the original variables $X$. More theoretical details on the nonparanormal family of distributions can be found in Liu et al. [32]. Additionally, to enforce sparsity, we assume the existence of a subset $S_0 \subset \{1, 2, \ldots, p\}$ with $|S_0| = O(1)$, such that, conditional on the features in $S_0$, the response $Y$ is independent of the features in $S_0^c$. In other words, $S_0 = \{k : f(y|X_k) \text{ depends on } X_k\}$, where $f(y|X)$ is the conditional density of $y$ given $X$. Our objective is to learn the sparsity structure by estimating the true index set $S_0$.

### 2.2 | Screening step

Assuming that the size of set $S_0$, $s$, is significantly smaller than the dimension of the feature space denoted as $p$, it follows that a large majority of the features reside outside of $S_0$, in $S_0^c$. Therefore, during the initial screening phase, our primary objective is to identify an active set $\hat{S}_n$ with a much smaller cardinality, denoted as $|\hat{S}_n| << p$. We aim for $P\left(S_0 \subset \hat{S}_n\right) \to 1$ as the sample size $n$ approaches infinity. This property, known as the “sure screening property” [17], guarantees the retention of all significant predictors within $\hat{S}_n$ with high probability, while effectively eliminating the other predictors $X_j, j \in \hat{S}_n^c$ from subsequent analysis. Considering that these active variables frequently display substantial intercorrelation due to their selection based on marginal correlation with the response variable, our subsequent step entails the utilization of their conditional dependency structure for clustering.

#### 2.2.1 | Finding the active predictors

To identify the active set effectively, we employ a two-step approach. First, we apply a nonparanormal transformation to the joint variables $(Y, X)$. This transformation aligns all variables as a joint Gaussian variable, preserving their conditional covariance structure. In the second step, we employ the Henze–Zirkler’s (HZ) test, which assesses significant dependence between the transformed response and predictors through pairwise testing. This approach, originally proposed by Liu et al. [32], Henze & Zirkler [22], and Xue & Liang [51], enables us to identify the active set while considering the complex relationships and dependencies among variables, making it a valuable tool in high-dimensional data analysis. The strategy proceeds as follows:

1. **Nonparanormal transformation:** We first consider the following transformation: $T_y(Y) = \Phi^{-1}(F_y(Y)), T_k(X_k) = \Phi^{-1}(F_k(X_k)), k = 1, 2, \ldots, p$, where $\Phi(\cdot)$ denotes the CDF of the standard Gaussian distribution. However, in practice, the cdf of $Y$ and $X_k$ are unknown, we can estimate it by the truncated empirical cdf as suggested by Liu et al. [32]. Henceforth, let $\left(\hat{T}_y(Y), \hat{T}_k(X_k)\right)$ denote the corresponding transformations.

2. **HZ test:** By the basic properties of CDF, it is easy to see that $\hat{(T_y(Y), T_k(X_k))}$ will jointly follow a bivariate Gaussian distribution $N_2(0, I_2)$ if and only if $Y$ is independent of $X_k$. This can be tested using HZ test, Henze & Zirkler [22], where the test statistic for the predictor $X_k$ can be expressed as $w_k = \int_{R^2} \psi_k(t) - \exp\left(-\frac{1}{2}t^t\right) \phi(t)dt, k = 1, 2, \ldots, p$, where $\psi_k(t)$ is the characteristic function of $\left(T_y(Y), T_k(X_k)\right)$ and $\exp\left(-\frac{1}{2}t^t\right)$ represents the characteristic function of $N_2(0, I_2)$. It typically measures the disparity between the joint distribution of $\left(T_y(Y), T_k(X_k)\right)$ and $N_2(0, I_2)$ and is expected to be typically high for the nonnull predictors $X_j, j \in S_0$ indicating significant evidence against the independence of the transformed variable $\left(T_y(Y), T_k(X_k)\right)$.
Next, as in practice, we proceed with \( \left( \tilde{T}_Y(Y), \tilde{T}_k(X_k) \right) \), we calculate the HZ test statistic as

\[
\tilde{w}_k = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \exp \left( \frac{-d_{ij}^2}{\tau^2} \right) - \frac{2}{n(1 + \beta^2)} \sum_{i=1}^{n} \exp \left( \frac{-d_{ii}^2}{(1 + \tau^2)^2} \right) + \frac{1}{1 + 2\beta^2},
\]

where \( d_{ij} = \left( \tilde{T}_k(x_{ki}) - \tilde{T}_k(x_{kj}) \right)^2 + \left( \tilde{T}_Y(y_i) - \tilde{T}_Y(y_j) \right)^2 \) and \( d_i = \tilde{T}_k(x_{ki}) + \tilde{T}_Y(y_i) \). Consistent with the existing literature, we choose the value of the smoothing parameter \( \beta \) as \( \frac{\sqrt{6n \log(n)}}{\sqrt{2}} \), which corresponds to the optimal bandwidth for a nonparametric kernel density estimator with Gaussian kernel [22]. The observed test statistics \( \tilde{w}_k \) converges to \( w_k \) as shown in Xue and Liang [51].

3. Next, we select the active set of predictors \( \hat{S}_n \) according to the larger values of \( \tilde{w}_k \), that is, \( \hat{S}_n = \{ 1 \leq k \leq p : \tilde{w}_k > cn^{-\kappa} \} \) where \( c \) and \( \kappa \) are predetermined threshold values.

The active set, denoted as \( \hat{S}_n \), comprises predictors that exhibit significant marginal correlation with the response variable. Importantly, under mild regularity conditions concerning the signal strength of the nonnull predictors, specifically when \( \min_{k \in \hat{S}_n} w_k \geq 2cn^{-\kappa} \) with \( c \) as a constant and \( 0 \leq \kappa \leq \frac{1}{2} \), the screening process demonstrates the coveted property of “sure screening” [51]. This property signifies that as the sample size \( n \) approaches infinity, the probability that the true active set \( S_0 \) is entirely contained within \( \hat{S}_n \) tends to unity.

Typically, practitioners set the active set size \( |\hat{S}_n| \) to a value like \( \nu_n = [n / \log(n)] \). Nevertheless, as we proceed to cluster the active variables in the subsequent step, our proposed approach exhibits robustness concerning the choice of \( |\hat{S}_n| \). In essence, it is crucial to retain most of the significant variables, and thus, we advocate selecting a larger active set size proportional to \( \nu_n \).

### 2.2.2 Clustering the active predictors using the precision matrix

The Henze–Zirkler test primarily evaluates pairwise marginal correlations, which may encompass null predictors displaying robust associations with significant predictors. This often leads to a situation where these null predictors exhibit substantial intercorrelations. In response to this challenge, our approach aims to mitigate the issue of excessive correlation within the active set. We achieve this by leveraging the conditional dependency structure and partitioning the active variables, denoted as \( X_j : j \in \hat{S}_n \), into a set of nonoverlapping clusters \( C_1, C_2, \ldots, C_p \), where \( p_c << p \).

In the context of high-dimensional feature spaces, the adoption of a sparse precision matrix, which excels at capturing the intricate dependence structure, has garnered considerable recognition within statistical literature (e.g., Lauritzen [25] and Shojaie & Michailidis [42]). This widespread acknowledgment stems from its scalability and effectiveness in modeling complex dependencies. Analyzing conditional dependencies unlocks invaluable insights that extend beyond the scope of what can be gleaned from a straightforward examination of covariance matrices. To illustrate, consider the study of the human brain: two distinct brain regions might exhibit a high correlation not due to a direct anatomical connection but rather owing to their robust interactions with a common third region. Consequently, neglecting these conditional dependencies and relying solely on correlation-based clustering would result in unwieldy cluster sizes and less interpretable groupings of correlated predictors, as exemplified in the context of brain region analysis [11].

To estimate the precision matrix, we employ the node-wise Lasso algorithm [47] on the transformed variables \( \left( \tilde{T}_Y(Y), \tilde{T}_k(X_k) \right) \). The node-wise Lasso regression is a common approach used to estimate a sparse precision matrix within the Gaussian graphical model framework. It achieves this by performing simultaneous Lasso regression on each predictor. The tuning parameters for each node-wise Lasso regression are typically chosen through cross-validation. For comprehensive insights into this algorithm and its associated theoretical guarantees, readers are referred to [35]. The estimated precision matrix by the node-wise Lasso algorithm is denoted as \( \hat{\Sigma}^{-1} \), while \( \rho(Z, Z') \) represents any correlation metric for two random variables \( Z \) and \( Z' \), such as Pearson’s correlation. Algorithm 1 provides a concise summary of the clustering step.

Here, our initial step involves the grouping of active predictors that exhibit conditional dependence on a specific predictor, denoted as \( X_i \in \hat{S}_n \), into distinct clusters labeled as \( C_i \). While clustering based on conditional dependence tends to yield smaller clusters, some overlaps may arise due to the intricate associations within the original predictor space. To mitigate excessive intercluster correlation, we consolidate clusters that exhibit a maximum correlation exceeding a predetermined threshold \( r \) (typically set to 0.9). Following this consolidation, each cluster undergoes refinement by integrating all other features that exhibit conditional dependence on existing cluster members.
In the clustering step, our primary objective extends beyond merely grouping the active predictors; we also aim to select a suitable representative from each cluster. To identify these representatives, we direct our attention to the HZ test statistic $\tilde{w}_i$ as described in 1. This statistic quantifies the similarity between the distribution of each nonparanormally transformed variable within a cluster and the null distribution $N(0, I_2)$. For each cluster $C_i$, we designate the variable $R_i = \arg \max_{j \in C_i} \tilde{w}_i$ as the representative. This choice signifies its strong association with the response variable when compared with other predictors within the same cluster.

2.3 | Cleaning with DNN

We start the cleaning step by modeling the response $Y$ and the cluster representatives $X_{\tilde{S}_n}$ obtained through Section 2.2.2. In order to perform the error-controlled variable selection, each representative will be assigned an importance score followed by a resampling algorithm to finally control the FDR.

While it is possible to adopt any other generic sparsity-inducing DNN procedure, here we focus on the LassoNet algorithm recently proposed by Lemhadri et al. [28] for its elegant mathematical frameworks which naturally sets the stage for nonlinear feature selection. To approximate the unknown functional connection, it considers the class of all fully connected feed-forward residual neural networks, namely, $\mathcal{F} = \{ f \equiv f_{\theta, W} : x \mapsto \theta^T x + h_W(x) \}$. Here, $W$ denotes the network parameters, $K$ denotes the size of the first hidden layer, $W^{(0)} \in \mathbb{R}^{p \times K}$ denotes the first hidden layer parameters, and $\theta \in \mathbb{R}^p$ denotes the residual layer's weights. In order to minimize the reconstruction error: the LassoNet objective function can be formulated as

$$\min_{\theta, W} L(\theta, W) + \lambda \|\theta\|_1 \text{ subject to } \left\| W_j^{(0)} \right\|_{\infty} \leq M |\theta|_j, \quad j = 1, 2, \ldots, p.$$  

With $L(\theta, W) = \frac{1}{n} \sum_{i=1}^{n} l(f_{\theta, W}(x_i), y_i)$ as the empirical loss on the training data and $x_i$ as the vector of cluster representatives observed for the $i$th individual. While the main feature sparsity is induced by the $L_1$ norm on residual layer parameter $\theta$, the second constraint controls the total amount of nonlinearity of the predictors. As mentioned in Lemhadri et al. [28], LassoNet can be argued as an extension of the celebrated Lasso algorithm to nonlinear variable selection.

In $L_1$ penalization framework, the importance of a specific feature is naturally embedded into the highest penalization level up to which it can survive in the model. So, to measure the importance of each representative, the LassoNet algorithm is executed over a long range of tuning parameter $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_r$ on $(Y, X_{\tilde{S}_n})$. In practice, a small value is fixed for $\lambda_1$ where all the variables are present in the model. Then, we gradually increase the value of the tuning parameter and stop at $\lambda_r$, where no variables are present in the model. Next, the importance score for the $j$-th cluster is defined as $\hat{\lambda}_j = \max_{\lambda \leq \lambda_r} \lambda$ up to which the $j$-th representative exists in the model, and then the following rank statistic is
computed: \( I_j = \sum_{j' \neq j} 1(\hat{\lambda}_{j'} \leq \hat{\lambda}_j) \) for \( j = 1, 2, \ldots, p_c \). A lower \( I_j \) means that the \( j \)-th cluster representative stays in the model up to a higher value \( \lambda \) implying its high potential as a significant cluster. In contrast, a higher \( I_j \) indicates the corresponding cluster leaves the model even for a smaller value of \( \lambda \) as a consequence of being simply a collection of null features. Hence, we should only focus on the clusters with lower ranks. Additionally, in order to control the FDR, understanding the behavior of the predictors under the null distribution is important. In traditional FDR-controlling algorithms, this is typically done by generating the \( p \)-values. Here, as a \( p \)-value-free algorithm, we propose the following resampling-based approach:

1. Generate \( B \) bootstrap versions of the data \( \{Y_b, X_b^S_h\}_{b=1}^B \) considering only the cluster representatives \( S_n \). For each bootstrap version, run the LassoNet algorithm parallelly, and calculate the importance of each representative by measuring \( \hat{\lambda}_b^j = \text{maximum value of } \lambda \text{ up to which the } j \text{-th predictor exists in the model for } b \text{-th bootstrap version} \), and then the ranks \( I_j^b = \sum_{j' \neq j} 1(\hat{\lambda}_{j'}^b \leq \hat{\lambda}_j^b) \).

Therefore, the averaged rank is \( \bar{I}_j = \frac{1}{B} \sum_{b=1}^B I_j^b \).
2. For an arbitrary threshold \( \delta \), we would select the cluster representatives with averaged rank \( \bar{I}_j \) lower than \( \delta \); so we define, \( N_+(\delta) = \sum_{j \in \bar{S}_n} 1(\bar{I}_j \leq \delta) \), representing the number of selected clusters with respect to the cutoff \( \delta \).
3. Next, to estimate the expected number of falsely discovered clusters, define \( R_b^\delta = \{ j : I_j^b \leq \delta \} \), the number of cluster representatives with higher importance score so that the corresponding rank is lower than the cutoff \( \delta \) in the \( b \)-th bootstrap version.

Additionally, define a neighborhood \( \mathcal{N}(\bar{I}_j, \kappa) = \{ l \in \{1, 2, \ldots, p_c \} : |\bar{I}_j - l| \leq \kappa \} \), for some specific small number \( \kappa \).

Further, we estimate the number of falsely discovered clusters and hence an estimator of the FDR can be constructed as \( \hat{\text{FDR}}(\delta) = \frac{\hat{N}_+(\delta)}{N_+(\delta)} \), where

\[
\hat{N}_+(\delta) = \frac{2B}{B} \sum_{b=1}^B \sum_{j \in R^b} 1(\bar{I}_j^b \in \mathcal{N}(\bar{I}_j, \kappa)) \tag{3}
\]

4. The \( \hat{\text{FDR}} \) is sequentially estimated with \( \delta = \bar{I}_{(1)}, \bar{I}_{(2)}, \ldots, \bar{I}_{(p_c)} \) and the optimum threshold is

\[
\Delta^* = \max\{ \delta > 0 : \hat{\text{FDR}}(\delta) < q \},
\]

for some pre-specific FDR-control level \( q \). The final selected set of clusters with controlled FDR is given by

\[
\tilde{D}_n = \{ C_j, j = 1, 2, \ldots, p_c : \bar{I}_j \leq \Delta^* \}.
\]

A visual representation of the entire algorithm is presented in Figure 3. The proposed method integrates the concept of false discovery through resampling. If a null predictor obtains a relatively high importance score, it is likely due to a specific bootstrap iteration that generates a spurious relationship, which is not consistent across all other bootstraps. Conversely, significant predictors should consistently yield higher importance scores across all bootstrap iterations. Consequently, the variability in importance score rankings will be substantially greater for null predictors compared with their nonnull counterparts. This principle, introduced in the statistical literature as bagging methods [4, 6], aims to reduce variance in black-box models for prediction/estimation. The proposed method leverages this phase transition phenomenon in the feature selection framework to effectively identify false discoveries. An empirical illustration of this phenomenon is presented in Figure 4. Although we employ nonparametric bootstrap for simplicity in this demonstration, other bootstrap techniques, such as multiplier bootstrap or perturbation bootstrap, can be readily applied. The theoretical investigation is provided in Appendix C, where, following the approach of perturbation bootstrap [36], we utilize random-weighted Group Lasso penalization to emulate the resampling setup.

2.3.1 How to choose the hyperparameter \( \kappa \)

Choosing an appropriate value of \( \kappa \) has a significant effect on the performance of SciDNet. A higher value of \( \kappa \) might lead to weaker control over the inclusion of false discoveries, whereas choosing a small \( \kappa \) will create tighter error control resulting in reduced power. However, we propose an effective way to tune the \( \kappa \) with the assistance of phase transition in the ranks of the importance score \( \bar{I}_j \) of the cluster representatives. For an illustration, in Figure 4, we consider a single index model (see Section 4 for more details of the data-generating mechanism), and the features with top 15 importance scores are shown along the \( x \)-axis. The first five representative features are the only relevant predictors (indicated by the vertical dotted red line). Along the \( y \)-axis, the center of the ellipse for each feature represents the rank of the importance scores \( \bar{I}_j \) averaged over 50 bootstrap replications and the area of each ellipse represents the bootstrap variability around the averaged
FIGURE 3 SciDNet architecture overview. The schematic diagram illustrates the proposed method SciDNet, where the green and red modules represent the screening and cleaning blocks, respectively.

FIGURE 4 Illustration of phase transition using synthetic data: features selected by SciDNet at $q = 0.2$ clearly have lower bootstrap variability compared with the other irrelevant features.

score. One would observe a clear phase transition in the bootstrap distribution of the ranks. For the significant features, the ranks are lower with extremely precise estimates. On the other hand, for the rest of the null features, the averaged ranks possess much higher values coupled with huge variability producing bigger ellipses. Hence, for a compact neighborhood $\mathcal{N}(\bar{T}_j, \kappa)$ to capture only the small variability in the bootstrap ranks of the significant features, we simply fix $\kappa = K^*$ (in Figure 4), the phase transition point for the averaged rank. This phase transition property is further illustrated on a real data in Appendix B.2.

3 | THEORETICAL DISCUSSION

Although the proposed method SciDNet is demonstrated based on the LassoNet algorithm [28], any other sparsity-inducing Deep learning framework can be adopted at the cleaning step of SciDNet. Hence, for the theoretical study, we consider a broader framework with a general analytic DNN. Also, as the screening step is theoretically guaranteed by Xue and Liang [51], we start the theoretical study directly from the cleaning step, assuming the data are $\{Y^i \in \mathbb{R}, X^i \in \mathcal{X}^p \}_{i=1}^n \sim P_D$, where $\mathcal{X}$ is a bounded open set in $\mathbb{R}^p$, $p < [n/\log(n)] < n$ is the dimension of the cluster representatives and the input density...
\( p_c \) is positive and continuous on its domain \( \mathcal{X} \). In the spirit of Dinh and Ho [14], we consider the following general analytic neural network model \( f_n(x) \), an L-layer neural network with parameters \( \alpha = (t_{in}, T_{in}, t_{out}, T_{out}, S) \) and defined by

- **Input layer:** \( h_1(x) = t_{in} + T_{in}x \)
- **Hidden layers:** \( h_j(x) = \phi_{j-1}(S, h_{j-1}(x), h_{j-2}(x), \ldots, h_1(x)), j = 2, 3, \ldots, L - 1 \)
- **Output layers:** \( f_n(x) = h_L(x) = t_{out} + T_{out}h_{L-1}(x) \)

with \( d_i = \text{size of the } i\text{-th layer}, d_1 = p, d_L = 1, T_{in} \in \mathbb{R}^{d_1 \times p}, t_{in} \in \mathbb{R}^{d_1}, T_{out} \in \mathbb{R}^{1 \times d_L}, t_{out} \in \mathbb{R}, \) and \( \phi_1, \phi_2, \ldots, \phi_{L-2} \) are analytic functions parameterized by the hidden layers’ parameter \( S \). This general framework covers a wide range of models, including feed-forward networks, convolutional networks, and a major subclass of residual networks. For the sake of theoretical study, we further assume that (1) the set of all feasible vectors \( \alpha \) of the model is a hypercube \( \mathcal{W} = [-A, A]^{p_c} \), (2) both \( \mathcal{W} \) and \( \mathcal{X} \) are bounded, (3) \( f_n \) is analytic in the sense that there exist \( C_1, C_2 > 0 \) such that \( |f_n(x)| \leq C_1 \), and \( \| \nabla f_n(x) \|_\infty \leq C_2, \forall x \in \mathcal{W}, x \in \mathcal{X} \) and these functions are Lipschitz continuous, and (4) \( X = f_n(X) + \epsilon \) with \( \epsilon \sim N(0, \sigma^2) \), \( \alpha^* \in \mathcal{W} = [-A, A]^{p_c} \) and we assume that the “true” model \( f_n(\cdot) \) only depends on \( x \) through a subset of significant features \( S \in 1, 2, \ldots, p \) while being independent of features in \( S^c, |S| = s = O(1) \). To this end, a general group Lasso estimator [14, 19] has been defined as

\[
\hat{\alpha}_n = \arg \min_{\alpha} \left\{ \frac{1}{n} \sum_{i=1}^{n} l(\alpha, X^i, Y^i) + \lambda_n \sum_{j=1}^{p} \|a^{\cdot, j}\| \right\},
\]

where \( l(\alpha, x, y) = (y - f_n(x))^2 \) is the square-error loss, \( \lambda_n > 0 \) is the associated penalty parameter, \( \| \cdot \| \) is the standard Euclidean norm and \( a^{\cdot, j} \) is the vector of parameters associated with \( j\text{-th input feature.} \)

Now, we incorporate the notion of resampling/bootstrap by introducing the random-weighted group lasso defined as follows:

\[
\hat{\alpha}_n^w = \arg \min_{\alpha} \left\{ \frac{1}{n} \sum_{i=1}^{n} W_i l(\alpha, X^i, Y^i) + \lambda_n \sum_{j=1}^{p} \|a^{\cdot, j}\| \right\},
\]

where \( W_i \overset{iid}{\sim} F_w \), independent of the data distribution \( P_D \). The random-weighting scheme effectively maintains the flavor of perturbation bootstrap [12] and thus can be used for uncertainty quantification in the context of sparse models. Ng & Newton [36] studied this in the context of sparse linear model. Here, we first show that under appropriate random weights, \( \hat{\alpha}_n^w \) converges to a set of well-behaved optimal hypotheses in the sense that \( \mathcal{K} = \{ a \in \mathcal{W} : f_a = f_{a^*} \text{ and } a^{\cdot, j} = 0, \text{for } j \in S^c \} \).

We note that, with the addition of random weights the overall probability measure changed to \( P = P_D \times P_w \) where \( P_w \) is the probability measure of the triangular array of random weights. We define the following three sigma fields: \( F_n^{w} = \sigma \{ W_1, W_2, \ldots, W_n \}, F_n^{h} = \sigma \{ X_1, X_2, \ldots, X_n \}, \) and \( F_n^{p} = \sigma \{ Y_1, Y_2, \ldots, Y_n \} \).

We further define, the risk function \( R(\alpha) = \mathbb{E}_{F_n^{w}}(Y - f_{\alpha}(X))^2, \) the empirical risk function \( R_n(\alpha) = \frac{1}{n} \sum_{i=1}^{n} (Y^i - f_{\alpha}(X^i))^2, \) and the weighted empirical risk function \( R_n^w(\alpha) = \frac{1}{n} \sum_{i=1}^{n} W_i (Y^i - f_{\alpha}(X^i))^2 \). The equivalence class can be expressed as \( H^* = \{ a \in \mathcal{W} : R(\alpha) = R(\alpha^* \} \).

For simplicity, in order to obtain a bounded weight, we assume \( F_w \) is such that \( P_w(C_2 < W < C_3) = 1, \) for some \( C_3 > 0 \).

**Lemma 3.1. (Probabilistic Lipschitzness of the random-weighted empirical risk).** For any \( \delta > 0, \) there exists \( M_\delta > c_0 \) such that \( R_n^w(\alpha) \) is an \( M_\delta \)-Lipschitz function with probability at least \( 1 - \delta \).

**Proof.** We note that

\[
|R(\alpha) - R(\beta)| = |\mathbb{E}\{(Y - f_{\alpha}(X))^2 - (Y - f_{\beta}(X))^2\}| \\
\leq E|\{f_{\alpha}(X) - f_{\beta}(X)\}(2Y - f_{\alpha}(X) - f_{\beta}(X))| \\
\leq C_2 \|\alpha - \beta\| E|\{2Y - f_{\alpha}(X) - f_{\beta}(X))| \\
\leq C_2 \|\alpha - \beta\| (2\sigma + 4C_1)
\]

Similarly,

\[
|R_n^w(\alpha) - R_n^w(\beta)| = \frac{1}{n} \sum_{i=1}^{n} W_i \{ (Y^i - f_{\alpha}(X^i))^2 - (Y^i - f_{\beta}(X^i))^2\} \\
\leq \frac{1}{n} \sum_{i=1}^{n} W_i |f_{\alpha}(X^i) - f_{\beta}(X^i))| \\
\leq \frac{1}{n} \sum_{i=1}^{n} W_i |2Y^i - f_{\alpha}(X^i) - f_{\beta}(X^i)| \\
\leq C_2 \|\alpha - \beta\| \frac{1}{n} \sum_{i=1}^{n} W_i |2Y^i - f_{\alpha}(X^i) - f_{\beta}(X^i)| \\
\leq C_2 \|\alpha - \beta\| (4c_1 + \frac{2}{n} \sum_{i=1}^{n} W_i |e^i|)
\]
Thus, for all $M_0 > 4C_1C_2$, the proof is completed by noting the following:

$$P\left( | R_n^w(\alpha) - R_n^w(\beta) | \leq M_0, \forall \alpha, \beta \in W \right)$$

$$\geq P\left( \frac{1}{n} \sum_{i=1}^{n} W_i | e_i | \leq \frac{M_0}{2C_1} - 2C_1 \right)$$

$$= E_{P_n}\left[ \frac{1}{n} \sum_{i=1}^{n} W_i | e_i | \leq \frac{M_0}{2C_1} - 2C_1 \right] - 2C_1| W_1, W_2, \ldots, W_n |$$

$$= E_{P_n}\left[ 1 - P\left( \frac{1}{n} \sum_{i=1}^{n} W_i | e_i | \geq \frac{M_0}{2C_1} - 2C_1 \right) \right]$$

$$\geq E_{P_n}\left[ 1 - \frac{\sum_{i=1}^{n} W_i | e_i |}{\frac{M_0}{2C_1} - 2C_1} \right] \geq 1 - \frac{C_1 | e_i |}{2C_1 - 2C_1}$$

Lemma 3.2. (Generalization bound for the random-weighted empirical risk). For any $\delta > 0$, $3C_1(\delta) > 0$ such that $P\left( | R_n^w(\alpha) - R(\alpha) | \leq C_4 \frac{\log n}{\sqrt{n}} \right) \geq 1 - \delta$, $\forall \alpha \in W$.

The rest of the proof follows the direct proof of the Lemma 3.3 in Dinh & Ho [14].

Theorem 3.3. (Convergence of random-weighted Group Lasso). For any $\delta > 0$, there exist $C_\delta, C' > 0$ and $N_\delta > 0$ such that for all $n \geq N_\delta$,

$$d\left( \tilde{\alpha}_n^w, H^* \right) \leq C_\delta \left( \frac{\lambda_n}{\sqrt{n}} + \frac{\log n}{\sqrt{n}} \right)$$

$$\leq 4C_4 \frac{\log n}{\lambda_n \sqrt{n}} + C'd\left( \tilde{\alpha}_n, H^* \right)$$

where $d(x, Z) = \inf_{z \in Z} \| x - z \|$.

The proof directly follows from the above-mentioned Lemma 3.1, 3.2 and the Theorem 3.3 from Dinh & Ho [14].

Theorem 3.1 demonstrates the convergence of the random-weighted group-Lasso estimates to a set of “well-behaved” optimal hypotheses $\mathcal{K} = \left\{ \alpha \in W : f_n = f_\alpha \right\}$ and $\left\| a^{1:2} \right\| = 0 \subset H^*$. However, this depends on the regularization parameter $\lambda_n$; finding its optimum value is generally a daunting task in practice. As a solution, the importance score considering the whole regularization path would provide a more robust way to identify false discoveries. Following our proposed method in Section 2.3, here we consider the cleaning step in view of the random-weighted group-Lasso penalized DNN. We repeat this step over $B$ independently generated random-weighting scheme and recall the importance scores $\hat{\lambda}_j$ and the corresponding ranks of the importance score $I_j$ for the $b$-th scheme as follows:

$$\hat{\lambda}_j^b = \max \left\{ \lambda_n \in W : \| a^{1:j}_n \| \neq 0 \right\}$$

and the averaged rank $\tilde{I}_j = \frac{1}{B} \sum_{b=1}^{B} I_j^b$.

Our basic strategy is to select the features with high importance scores consistent in all the bootstrap replications. Hence, for a cutoff $\delta$, define the number of selected features as $N_\delta(\delta) = \sum_{j=1}^{B} 1 \left( \tilde{I}_j \leq \hat{\lambda}_j^b \right)$. We further propose our estimate of the FDR as

$$\tilde{FDR}(\delta) = \frac{\hat{\alpha}_0(\delta)}{N_\delta(\delta)}$$

where $\hat{\alpha}_0(\delta) = \text{estimated number of false discoveries} = \sum_{b=1}^{B} \left\{ \sum_{j=1}^{B} 1 \left( I_j^b \leq \delta, I_j^b \in \mathcal{N}(\tilde{I}_j^b, K) \right) \right\}$. We further calculate the data-dependent optimum cut-off as $\Delta^* = \max \left\{ \delta \in \left\{ \tilde{I}_1, \tilde{I}_2, \ldots, \tilde{I}_B \right\} : \tilde{FDR}(\delta) < q \right\}$ for some pre-specific FDR-control level $q$.

Theorem 3.4. (Convergence of the estimated FDR). Using the data-dependent cut-off $\Delta^*$, the actual FDR is bounded by the
user-specified level \( q \); that is \( E\left( \frac{\hat{\alpha}(\hat{\Delta}^*)}{N_{\hat{\Delta}^*}} \right) < q \) as \( n \to \infty \).

**Proof.** The feature importance scores \( \hat{T}_j \) evidently provide the information of the survival of the feature \( X_j \) over the whole regularization path. Now, under the setup of the random-weighted group-lasso framework, this is uniquely monitored by the KKT condition: \( \|a^{(L)}\|_0 = 0 \) if \( \frac{d}{dx}(\sum_j a_{ij}^T)\) \( p_j \times n \) \( \text{diag}(W_1, W_2, \ldots, W_n) \) \( (Y - f_a(x))_{p \times n} \) \( < 1 \) \( \sum_j \hat{\lambda}_j \leq \hat{\lambda}_{ij} \) \( \hat{\lambda}_j \) \( \hat{\lambda}_j \) \( \sum_{j \neq j'} \left( \hat{\lambda}_j \leq \hat{\lambda}_{j'} \right) \sim U(p - s + 1, p) \). Now, we consider the following random variable named FDP whose expectation is the FDR.

\[
\text{FDP} = \frac{\sum_{j \in S_0^c} \{1 \mid \hat{T}_j \leq \hat{\Delta}^*, \hat{T}_j \in \mathcal{N}(\hat{\Delta}^*, \kappa)\}}{\sum_{j \in S_0} \{1 \mid \hat{T}_j \leq \hat{\Delta}^*\}} \leq q
\]

\[
\leq \frac{\sum_{j \in S_0^c} \{1 \mid \hat{T}_j \leq \hat{\Delta}^*, \hat{T}_j \in \mathcal{N}(\hat{\Delta}^*, \kappa)\}}{\sum_{j \in S_0} \{1 \mid \hat{T}_j \leq \hat{\Delta}^*\}} \leq \frac{1}{R(\Delta^*)}
\]

\[
\text{lim}_{n \to \infty} E(R(\Delta^*)) \leq 1.
\]

\[
E(R(\Delta^*)) = E\left( \frac{\sum_{j \in S_0} \{1 \mid \hat{T}_j \leq \hat{\Delta}^*, \hat{T}_j \in \mathcal{N}(\hat{\Delta}^*, \kappa)\}}{\sum_{j \in S_0} \{1 \mid \hat{T}_j \leq \hat{\Delta}^*\}} \right)
\]

\[
= \frac{\sum_{j \in S_0} \{1 \mid \hat{T}_j \leq \hat{\Delta}^*, \hat{T}_j \in \mathcal{N}(\hat{\Delta}^*, \kappa)\}}{\sum_{j \in S_0} \{1 \mid \hat{T}_j \leq \hat{\Delta}^*\}} \leq \frac{1}{R(\Delta^*)}
\]

\[
= \frac{\sum_{j \in S_0} \{1 \mid \hat{T}_j \leq \hat{\Delta}^*, \hat{T}_j \in \mathcal{N}(\hat{\Delta}^*, \kappa)\}}{\sum_{j \in S_0} \{1 \mid \hat{T}_j \leq \hat{\Delta}^*\}} \leq \frac{1}{R(\Delta^*)}
\]

The first part of FPR is \( \leq q \) by the typical choice of \( \Delta^* \) and hence, in order to show the asymptotic FDR control, all we need to show is

**4 | NUMERICAL ILLUSTRATIONS**

In this section, we investigate the finite-sample performance of SciDNet using a wide spectrum of simulation scenarios. We compare SciDNet with several baseline methods which are widely used in practice. We conduct this simulation study on synthetic data generated from a high-dimensional regression problem.

**4.1 | Data generation**

We first consider the single index model for the data-generating mechanism, which is a straightforward yet flexible example of nonlinear models. Here the response is related to a linear combination of the features...
through an unknown nonlinear, monotonic link function, i.e., \( y = g(x' \beta) + \epsilon \). We choose the following link functions: \( g(x) = \frac{x^2}{10} + \frac{3x}{10} \).

We set \( n = 400 \) and \( p = 5000 \). The coefficients \( \beta \in \mathbb{R}^p \) is sparse with the true nonzero locations \( S_0 = \{50, 150, 250, 350, 450\} \), \( s = |S_0| = 5 \), where \( \beta_{S_0} = 0 \), \( \beta_{S_0} \sim N_5(u \beta_0, 0.1I_{5 \times 5}) \) with \( u = (\pm 1)^5 \). The value of \( \beta_0 \) is set at \( \beta_0 = 2.4 \) to incorporate varying signal strength. The random error \( \epsilon \sim N(0, \sigma^2) \), with three increasing noise level as \( \sigma^2 = 1, 5, 10 \). The high-dimensional predictors are generated from \( X \sim N_p(0, \Sigma) \) where the covariance matrix \( \Sigma \) is chosen as a Toeplitz matrix with \( \Sigma_{ij} = \rho^{|i-j|} \). To check the effect of multicollinearity, we consider three different settings: \( \rho = 0.1, 0.5, 0.95 \).

### 4.2 Evaluation metrics

Let \( \hat{D}_q \) denote the selected set of features by some algorithm; then, we use the following three metrics to evaluate the performance of these feature selection algorithms:

1. **Power** = \( \frac{|\hat{D}_q \cap S_0|}{|S_0|} \), the proportion of relevant features that are correctly identified
2. **Empirical FDR** = \( \frac{|\hat{D}_q \cap S_0|}{|\hat{D}_q|} \), the proportion of falsely identified features among all the identified features
3. **n_{var}** = \( |\hat{D}_q| \), the number of total features selected and \( n_{clust} = |S_0| \), the number of clusters selected by a group feature selection method like SciDnet.

The whole experiment is repeated over 100 Monte Carlo replications and we summarize the results from the empirical evaluations in the following three subsections: (1) The effect of multicollinearity on major existing feature selection methods, (2) the Power vs. FDR balance of SciDNet, and (3) an ablation study showing the effectiveness of all the steps for SciDNet.

### 4.3 The effect of multicollinearity on major existing feature selection methods

We present a simulation study to evaluate the performance of our proposed algorithm in comparison with the following five baseline feature selection methods:

1. **Lasso** [46]: the \( L_1 \) penalized linear regression to prevent overfitting and improve model interpretability.
2. **Model-X knockoff** [8]: A theoretically guaranteed statistical method for FDR control used in high-dimensional variable selection. It constructs “knockoff” variables that mimic the correlations between the original variables and their relationship with the response variable, allowing for control of the false discovery FDR while identifying important variables.
3. **SurvNet** [44]: A DNN-based FDR-control method for feature selection applicable to high-dimensional large datasets.
4. **Deep Feature Selection (DFS)** [10]: A novel DNN method for feature selection in a high-dimensional setting with complex nonlinear relationships utilizing \( L_0 \) penalization.
5. **LassoNet** [28]: The nonlinear extension of Lasso. It combines the advantages of both \( L_1 \) penalization and neural network structures to identify the important features.

Although several other existing methods exist in the literature for ultrahigh-dimensional feature selection, we choose these five baseline methods because of their wide applicability and reliable theoretical guarantees. Also, these five methods can be thought of as representative of different classes of algorithms. For example, Model-X knockoff represents the big class of knockoff-based algorithms, whereas SurvNet, DFS, and LassoNet show the effectiveness of the DL-based feature selection methods. Among these five baseline methods, Model-X knockoff and SurvNet are designed to control the FDR and we use \( q = 0.2 \) as the FDR-control threshold. Also, Lasso, LassoNet, and DFS require proper tuning for their \( L_1 \) or \( L_0 \) penalty parameters, which can be done via a grid search. For this purpose, we optimize a BIC-type criterion [9] to tune these hyperparameters, as suggested by the authors of Chen et al. [10]. For the practical implementation of these baseline methods, we used the code/hyperparameters provided by the authors in the respective papers.

Figure 5 demonstrates the Power, FDR, and the \( n_{var} \) of all these methods under different correlation strengths and varying noise levels, fixing \( \beta_0 = 2 \). Also, as SciDNet selects the features as clusters, we show the \( n_{clust} \) in numbers along with \( n_{var} \), in the third row of Figure 5. One would observe that the baseline methods perform poorly as the autocorrelation increases; they result in reduced power and inflated false discoveries. The FDR-controlling methods such as Model-X knockoff and SurvNet fail to control their FDR under the prespecified threshold \( q = 0.2 \) for higher autocorrelation. This is somehow expected, as these methods are not tailored for handling such huge multicollinearity. This demonstration empirically motivates as well why we need a feature selection method designed for highly correlated feature space. The DL-based method like LassoNet, DFS, or Survnet additionally suffers from insufficient data under the current big-p-small-n setting. Our
additional experiments, presented in Appendix D, demonstrate how these DL-based models gain better power-FDR balance given sufficient training data and moderate correlation among the features. Compared with these baseline methods, SciDNet successfully maintains its power while controlling the FDR below the prespecified threshold \( q = 0.2 \) irrespective of the correlation strength. Also, under moderate multicollinearity, when there is no need for clustering, most of the selected clusters by SciDNet are just singleton sets. On the other hand, under the setting of excessive multicollinearity, the individual features become almost indistinguishable. SciDNet addresses this added uncertainty by selecting larger clusters for higher autocorrelation, as demonstrated in the third row of Figure 5.

4.4 The power versus FDR balance of SciDNet

One major validation for an FDR-controlling method is to check how it retrieves its power with respect to gradually increasing the FDR-control threshold \( q \). Figure 6 shows this power-FDR trade-off for SciDNet. Also, Section 4.1 demonstrates that the baseline methods are not well suited for highly correlated feature space. As SciDNet is specifically designed for this setting, for ease of illustration, we only present the performance of SciDNet setting \( \rho = 0.95 \). Figure 6 illustrates how our method enjoys quick recovery in power when we gradually increase the FDR-controlling threshold \( q \) from 0.01 to 0.20 and also maintain the number of false discoveries below the required level. This illustration empirically validates SciDNet as an FDR-controlled feature selection method.

4.5 An ablation study

The proposed method SciDNet is a multistep process. Its screening step reduces the dimension while retaining the main important features and clustering the highly correlated features. Following this, the cleaning step further uses a sparsity-inducing DL model and finally selects some cluster of features by controlling the FDR via resampling. We aim to analyze the impact of these two steps on the overall performance of the model by adding them one by one and observing the change in the evaluation metrics. We also compare our proposed cleaning step (i.e. resampled LassoNet with FDR estimation) with other possible alternatives like using knockoffs for cleaning.

Hence, for the ablation study, we compare the following four methods under varying signal and noise strength:
1. **Screening only**,  
2. **Screening + LassoNet**: Here we consider LassoNet as the cleaning step  
3. **Screening + Knockoff**: Here we consider Model-X knockoff as the cleaning step, with the FDR-controlling threshold $q = 0.2$  

**Screening + resampled LassoNet**, that is, SciDNet. We use two FDR-controlling thresholds $q = 0.1$ and 0.2.

The power, empirical FDR, $n_{\text{var}}$ are illustrated in Figure 7. In addition, the selected number of clusters, that is, $n_{\text{clust}}$, are written in numbers in the third row of Figure 7. It empirically consolidates several interesting characteristics: (a) due to the surescreening property, the screening step selects a slightly bigger set of features resulting in high power and high FDR which necessitates further cleaning; (b) all the alternative cleaning steps aim to eliminate the null features further and reduce the FDR while maintaining the power of the screening step; and (c) for higher signal and low noise case, SciDNet is comparable with other alternatives. However, for difficult scenarios like low signal and high noise case, which is common in modern genomic and imaging datasets, SciDNet maintains its performance. One would notice that Model-X knockoff loses its FDR control at the nominal level $q = 0.2$ and results in inflated FDR in the presence of high noise. On the other hand, by effectively using the added information from the resampling, SciDNet achieves the best performance (in terms of the power-FDR trade-off) and continues to preserve its FDR below the nominal level $q$.

Overall, our ablation study showed that the proposed cleaning following the screening step contributes to the overall performance of SciDNet and that removing any of them leads to a significant drop in accuracy. The screening helps in reducing the dimension of the feature space by removing a large chunk of irrelevant features, which further makes the cleaning step computationally very efficient. We also conducted a further simulation study considering several nonlinear models as data-generating processes with Gaussian and non-Gaussian features. The results are presented in Appendix C which further substantiates the results mentioned above that SciDNet maintains a satisfactory power-FDR balance for various complicated nonlinear models with and without interaction terms. The hyperparameter selection and further implementation details of SciDNet are relegated to Appendices B.1 and E, respectively.
4.6 Data-adaptive cluster formation with varying multicollinearity

An intriguing characteristic of SciDNet lies in its adaptive estimation of cluster size, enabling it to effectively address the additional uncertainty arising from significant multicollinearity. To demonstrate this capability, we employed SciDNet in a scenario where the data-generating mechanism followed a nonlinear additive model, represented as

\[ y = 2X_{50} + 0.5 \exp(X_{150}) + 2X_{250}^3 + 6X_{350} + 2\text{ReLU}(X_{450}) + \epsilon, \]

with \( \epsilon \sim N(0, \sigma^2) \). We set \( \sigma^2 = 5 \), while keeping the rest of the simulation setting consistent with the description at the beginning of Section 4. The autocorrelation coefficients among the features were varied as \( \rho = \{0.1, 0.2, \ldots, 0.9\} \).

Figure 8 demonstrates several key aspects of SciDNet’s performance. First, with varying levels of multicollinearity strength, SciDNet effectively maintains a desirable power-FDR trade-off, successfully controlling the FDR below the nominal level \( q = 20\% \). Second, SciDNet adapts to the added uncertainty caused by high multicollinearity by inflating the cluster size. Notably, in the figure, when there is lower autocorrelation and no need for clustering, the number of selected features (\( n_{var} \)) and the number of selected clusters (\( n_{clust} \)) are almost equal, indicating that the majority of selected clusters are single-tone sets. Conversely, for higher autocorrelation, SciDNet produces larger clusters, enhancing intra-block correlation and reducing inter-block correlation, while keeping the number of clusters (\( n_{clust} \)) nearly constant, thereby achieving a stable power-FDR balance across the entire range of the autocorrelation coefficient \( \rho \). This experiment provides additional evidence of the robustness of our proposed method.

Overall, the simulation study provides strong evidence supporting several key aspects of SciDNet, including its effectiveness, ablation, and robustness. Furthermore, due to the independent nature of most of the intermediate steps of SciDNet, parallel execution is possible, rendering it lightweight and computationally feasible even for ultrahigh-dimensional datasets. As an illustration, for this simulation setting, the complete execution of SciDNet takes approximately 1.56 min (61.31 s for screening and 32.57 s for cleaning), in contrast to the Model-X knockoff Candès et al. [8] which requires 7.4 min for generating knock-off features in a 2000-dimensional feature space using “SDP”-style knock-off generation (refer to Patterson & Sesia [37] for more details). Notably, SciDNet eliminates the need for hyperparameter grid search, which can be challenging in practice for ultrahigh-dimensional datasets. For instance, DFS [10] requires approximately 1.65 min for execution with a fixed hyperparameter value,
followed by an extensive grid search that inflates computational complexity. All experiments were performed on a high-performing computing facility equipped with Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz and 4 Tesla V100S. The source code is available at the repository (link: https://anonymous.4open.science/r/SciDNet-74D3).

5 | REAL DATA ANALYSIS

In addition to the simulation studies, we implemented the proposed algorithm SciDNet in the following two publicly available gene-expression data sets—the CCLE dataset and the riboflavin dataset. We substantiate the findings in two ways: We first provide supporting evidence from the domain research. Additionally, as a more data-aligned validation, we demonstrate that several generic prediction models significantly gain in test accuracy when applied only on the few features selected by SciDNet compared with the prediction result considering the whole feature space. For this purpose, consistent with the other genomic studies, we use the prediction correlation \( \text{Corr}(Y_{\text{Pred}}, Y_{\text{Test}}) \) in addition to the test MSE as a metric to measure the test performance. To overcome the extra burden of the low sample size and ultrahigh dimensionality in these data sets, we consider 50 independent replications where the data is divided into training and testing maintaining an 8 : 2 ratio to get the metrics for the test performance. The final estimate is obtained by averaging all the test MSEs calculated on each of these replications. A similar approach is considered for the correlation metric as well.

5.1 Selection of drug-sensitive genes using CCLE dataset

A recent large-scale pharmacogenomics study, namely, the cancer cell line encyclopedia (CCLE), investigated multiple anticancer drugs over hundreds of cell lines. Its main objective is to untangle the response mechanism of anticancer drugs which is critical to precision medicine. The data set consists of dose–response curves for 24 different drugs across over \( n = 400 \) cell lines. Each cell line consists of the expression data of \( p = 18,926 \) genes, which we consider as features. For the response, we used the area under the dose–response curve, known as activity area in [2], as the response variable to measure drug sensitivity. The activity area represents the extent to which a drug enters a person’s bloodstream after administration and is a continuous variable. Here, in this ultrahigh-dimensional regression setting, we seek to uncover the set of genes associated with the following five specific anticancer drug sensitivities: Topotecan, 17-AAG, Irinotecan, Paclitaxel, and AEW541. These drugs have been used to treat ovarian cancer, lung cancer, and other cancer types. Previous research outputs on these drugs and related gene-expression data can be found elsewhere [2].
SciDNet produces multi-resolutional clusters of genes for each of the five drugs considered which are interpretable from the domain science perspective. For example, SciDNet discovers \textit{SLFN11} as the top drug-sensitive gene for the drugs Topotecan and Irinotecan. This is consistent with the previous findings as Barretina et al. [2]; Zoppoli et al. [55] reported the gene \textit{SLFN11} to be highly predictive for both drugs. For another drug 17-AAG, SciDNet discovers the gene \textit{NQO1} as the top-most important gene which is known to be highly sensitive to 17-AAG [21]. The full table containing all the genes selected by SciDNet at \( q = 15\% \) error-control level and relevant findings from previous genomic studies have been relegated to Appendix F.

Additionally, as in a real setting, it is difficult to check the performance of a feature selection algorithm; here we present a more data-oriented statistical evaluation for further endorsement of SciDNet’s discoveries. From the prediction aspect, one would expect that a prediction model implemented only on a handful of features selected by a successful feature selection algorithm would maintain the similar performance of a model implemented on the whole feature space; in some cases, it might enhance the accuracy. To validate this, we randomly split the whole data into 8:2 for training and testing. First, the SciDNet is implemented in the training part, and then two separate prediction models are used only focusing on the selected features: (1) an MLP—a feed-forward multi-layer perceptron with two hidden layers and (2) bagged regression tree [4]. These two experiments are henceforth called: “SciDNet + MLP” and “SciDNet + RT”. Next, the test data are used to check the out-of-sample prediction accuracy. Furthermore, similar to the simulation study in Section 4, we separately implement the LassoNet on the training data for its simultaneous sparsity-induced prediction-optimal characteristics. The summary of the results is presented in Table 1 which indicates several interesting points. First, to get the prediction-optimal result, LassoNet fails to capture the sparsity and discovers a huge number of genes. This is somehow expected as most of the prediction-optimal sparse methods tend to select a larger set of features to maintain the prediction quality [49]. On the other hand, SciDNet produces only \( \sim 10 \) clusters with an average cluster size \( \sim 2.5 \). Even with this huge dimension reduction, the added gain in test MSE and \( \text{Corr}(Y_{\text{Pred}}, Y_{\text{Test}}) \) further proves that SciDNet successfully retains all the significant predictors. One would further notice that SciDNet + RT achieves the best stable performance which is consistent for all the drugs and our simulation study as well. This experiment demonstrates that a black-box predictive model produces more accurate results when applied on the features selected by SciDNet rather than its implementation on the whole feature space, indicating SciDNet’s potential use in both feature selection and prediction (Table 1).

### 5.2 Selection of associated genes in Riboflavin production data set

We further implement the SciDNet in the context of riboflavin (vitamin B2) production with bacillus subtilis data, a publicly accessible dataset available in the “hdi” package in R [13]. Here the continuous response is the logarithm of the riboflavin production rate, observed for \( n = 71 \) samples along with the logarithm of the expression level of \( p = 4088 \) genes which are treated here as the predictors. Unlike the previous CCLE data, significant multicollinearity is present in most of the Riboflavin data set, as demonstrated in Figure 9. Hence, to determine which genes are important for riboflavin production, SciDNet resulted in finding nine clusters of a total of 160 correlated genes at the \( q = 15\% \) FDR-control level, making the average cluster size of \( \sim 17.78 \), which is much bigger compared with the previous analysis. SciDNet discovered the gene \textit{YCIC} as one of the expressive genes related to riboflavin production which was identified by Bühlmann et al. [7] as a causal gene in this context. The full list of the selected cluster of genes by SciDNet is relegated to Appendix F.2.
FIGURE 9  A snapshot of correlation strength for first 100 genes considered for the drug Topotecan in CCLE dataset (left) and riboflavin dataset (right).

TABLE 2  Prediction performance of SciDNet for Riboflavin production data set.

| Algorithm               | Test MSE | Corr \( (Y_{\text{Pred}}, Y_{\text{Test}}) \) |
|-------------------------|----------|--------------------------------------------|
| LassoNet                | 0.83 (0.18) | 0.36 (0.30)                               |
| SciDNet + LassoNet      | 0.19 (0.15) | 0.89 (0.12)                               |
| SciDNet + RT            | 0.42 (0.16) | 0.74 (0.15)                               |
| SciDNet + MLP           | 2.64 (0.79) | 0.28 (0.37)                               |

The results from the empirical validation of SciDNet’s feature selection on the riboflavin dataset are presented in Table 2. However, for the empirical evaluation, SciDNet+MLP is performing poorly as the inflated cluster dimensions make the input layer of the MLP comparatively large where the number of the training data point is \( \approx 57 \). This necessitates the need for a sparse model here, and we adopt the idea of Relaxed Lasso, first proposed by Meinshausen [34]. Here we implement the LassoNet again on the selected features by SciDNet, which certainly improves the prediction accuracy. Consistent with the previous experiments, SciDNet+RT effectively maintains its prediction performance. This further consolidates the need for applying an apt feature selection method before fitting a predictive model for an explainable research outcome.

6 | DISCUSSION

While the explainable AI is the need of the hour, statistical models coupled with cutting-edge ML techniques have to push forward because of their solid theoretical foundation clipped with principled algorithmic advancement. The proposed method SciDNet efficiently exploits several existing statistics and ML literature tools to circumvent some of the complexities that simple adaptation of current DL-based models fail to address appropriately. The basic intuition and exciting empirical results of SciDNet on simulated and real datasets open avenues for further research. For example, one may be interested in developing a theoretical foundation for this “screening” and “cleaning” strategy for provable FDR control. It would be worth mentioning that although we used the sure independence screening with HZ test and LassoNet as the main tools, SciDNet puts forward a more generic framework and can be implemented with any other model-free feature screening method and sparsity-inducing DL algorithms like Feng & Simon [19]. In the screening part, a further methodological extension would consider relaxing the assumption of nonparanormally distributed features for a more flexible approach. Additionally, as the dimensionality is reduced after the screening step, it would be interesting to implement model-free knockoff generating algorithms like Romano et al. [38] in the cleaning step as further algorithmic development. One limitation is that we mainly focus on the regression setup with the continuous outcome because of the requirements of the HZ sure Independence test used in the screening step. For a classification task, any model-free feature screening method like Zhou & Zhu [54] can be applied in a more general framework.

FUNDING INFORMATION
This research has been supported by the National Science Foundation grant DMS1916339.
DATA AVAILABILITY STATEMENT
The data that support the findings of this study are openly available in CCLE - Cancer Cell Line Encyclopedia at https://sites.broadinstitute.org/ccle. Additionally, the Riboflavin dataset is available in the “hdi” package in R.

ORCID
Arkaprabha Ganguli https://orcid.org/0000-0002-5588-2157

REFERENCES
1. R. F. Barber, E. J. Candès, and R. J. Samworth, Robust inference with knockoffs, Ann. Stat. 48 (2020), no. 3, 1409–1431.
2. J. Barretina, G. Caponigro, N. Stranksy, K. Venkatesan, A. A. Margolin, S. Kim, C. J. Wilson, J. Lehár, G. V. Kryukov, D. Sonkin, A. Reddy, M. Liu, L. Murray, M. F. Berger, J. E. Monahan, P. Morais, J. Melzter, A. Korejwa, J. Jané-Valbuena, F. A. Mapa, J. Thibault, E. Bric-Furlong, P. Raman, A. Shipway, I. H. Engels, J. Cheng, G. K. Yu, J. Yu, P. Aspeli, M. de Silva, K. Jagtap, M. D. Jones, L. Wang, C. Hatton, E. Palescandolo, S. Gupta, S. Mahan, C. Sougnez, R. C. Onofrio, T. Liefeld, L. Mac-Conaill, W. Winckler, M. Reich, N. Li, J. P. Mesirov, S. B. Gabriel, G. Getz, K. Ardille, V. Chan, V. E. Myers, B. L. Weber, J. Porter, M. Warmuth, P. Finan, J. L. Harris, M. Meyerson, T. R. Golub, M. P. Morrissey, W. R. Sellers, R. Schlegel, and L. A. Garraway, The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity, Nature 483 (2012), no. 7391, 603–607.
3. Y. Benjamini and Y. Hochberg, Controlling the false discovery rate: A practical and powerful approach to multiple testing, J. R. Stat. Soc. Ser. B Methodol. 57 (1995), no. 1, 289–300.
4. L. Breiman, Bagging predictors, Mach. Learn. 24 (1996), no. 2, 123–140.
5. D. Brzyski, C. B. Peterson, P. Sobczyk, E. J. Candès, M. Bogdan, and C. Sabatti, Controlling the rate of GWAS false discoveries, Genetics 205 (2017), no. 1, 61–75. https://doi.org/10.1534/genetics.116.193987.
6. P. Bühlmann and Y. Bin, Analyzing bagging, Ann. Stat. 30 (2002), no. 4, 927–961. https://doi.org/10.1214/aos/1031689014.
7. P. Bühlmann, M. Kalisch, and L. Meier, High-dimensional statistics with a view toward applications in biology, Ann. Rev. Stat. Appl. 1 (2014), no. 1, 255–278.
8. E. J. Candès, Y. Fan, L. Janson, and J. Lv, Panning for gold: Model-x knockoffs for high-dimensional controlled variable selection, arXiv: Methodology 80 (2016), 551–577.
9. J. Chen and Z. Chen, Extended Bayesian information criteria for model selection with large model spaces, Biometrika 95 (2008), no. 3, 759–771. https://doi.org/10.1093/biomet/asn034.
10. Y. Chen, Q. Gao, F. Liang, and X. Wang, Nonlinear variable selection via deep neural networks, J. Comput. Graph. Stat. 30 (2021), no. 2, 484–492.
11. A. Das, A. L. Sampson, C. Lainscsek, L. Muller, W. Lin, J. C. Doyle, S. S. Cash, E. Halgren, and T. J. Sejnowski, Interpretation of the precision matrix and its application in estimating sparse brain connectivity from long sleep spindles from human electroencephalography recordings, Neural Comput. 29 (2017), no. 3, 603–642. https://doi.org/10.1162/NECO_a_00936.
12. D. Das and S. N. Lahiri, Distributional consistency of the lasso by perturbation bootstrap, Biometrika 106 (2019), no. 4, 957–964. https://doi.org/10.1093/biomet/asz029.
13. R. Dezeure, P. Bühlmann, L. Meier, and N. Meinshausen, High-dimensional inference: Confidence intervals, p-values and R-Software hdi, Statist. Sci. 30 (2015), no. 4, 533–558. https://doi.org/10.1214/15-STSS257.
14. V. C. Dinh and L. S. Ho, Consistent feature selection for analytic deep neural networks, Adv. Neural Inf. Proces. Syst. 33 (2020), 2420–2431.
15. S. N. Dorman, K. Baranova, J. H. M. Knoll, B. L. Urquhart, G. Mariani, M. L. Carcangi, and P. K. Rogan, Genomic signatures for paclitaxel and gemcitabine resistance in breast cancer derived by machine learning, Mol. Oncol. 10 (2016), no. 1, 85–100.
16. D. Elbrächter, D. Perekrestenko, P. Grohs, and H. Bölcskei, Deep neural network approximation theory, IEEE Trans. Inf. Theory 67 (2021), no. 5, 2581–2623.
17. J. Fan and J. Lv, Sure independence screening for ultrahigh dimensional feature space, J. Roy. Statis. Soc. Series B (Statist. Methodol.) 70 (2008), no. 5, 849–911.
18. J. Fan and J. Lv, A selective overview of variable selection in high dimensional feature space, Stat. Sin. 20 (2010), no. 1, 101–148.
19. J. Feng and N. Simon, Sparse-input neural networks for multivariate normality, Commun. Statis. Theory Methodol.) 81 (2019), no. 1, 45–74.
20. J. Fan and J. Lv, Identifying causal variants at loci with multiple signals of association, Genetics 198 (2014), no. 2, 497–508.
21. J. Jordon, Y. Chen, and M. van der Schaar. KnockoffGAN: Generating knockoffs for feature selection using generative adversarial networks. In International Conference on Learning Representations. 2019 https://openreview.net/forum?id=ByeZ5jC5YO.
22. S. L. Lauritzen, Graphical models, Vol 17, Oxford University Press, England, 1996.
23. H. J. Lee, M. Hanibuchi, S.-J. Kim, Y. Hyunkyung, M. S. Kim, J. He, R. R. Langley, F. Lehembre, U. Regenass, and I. J. Fidler, Treatment of experimental human breast cancer and lung cancer metastases in mice by macitentan, a dual antagonist of endothelin receptors, combined with paclitaxel, Neurol Oncology 3681–3688.
24. J. Feng and N. Simon, Sparse-input neural networks for multivariate normality, Commun. Statis. Theory Methodol.) 81 (2019), no. 1, 45–74.
25. F. Hormozdiari, E. Kostem, E. Y. Kang, B. Pasaniuc, and E. Eskin, Identifying causal variants at loci with multiple signals of association, Genetics 198 (2014), no. 2, 497–508.
26. J. Jordon, Y. Chen, and M. van der Schaar. KnockoffGAN: Generating knockoffs for feature selection using generative adversarial networks. In International Conference on Learning Representations. 2019 https://openreview.net/forum?id=ByeZ5jC5YO.
27. S. L. Lauritzen, Graphical models, Vol 17, Oxford University Press, England, 1996.
28. H. J. Lee, M. Hanibuchi, S.-J. Kim, Y. Hyunkyung, M. S. Kim, J. He, R. R. Langley, F. Lehembre, U. Regenass, and I. J. Fidler, Treatment of experimental human breast cancer and lung cancer metastases in mice by macitentan, a dual antagonist of endothelin receptors, combined with paclitaxel, Neurol Oncology 3681–3688.
29. A. Li and R. F. Barber, Multiple testing with the structure-adaptive benjamini-hochberg algorithm, J. Roy. Statis. Soc. Series B (Statist. Methodol.) 81 (2019), no. 1, 45–74.
Additional supporting information can be found online in the Supporting Information section at the end of this article.

**How to cite this article:** A. Ganguli, T. Maiti, and D. Todem, *Error-controlled feature selection for ultrahigh-dimensional and highly correlated feature space using deep learning*, Stat. Anal. Data Min.: ASA Data Sci. J. 17 (2024), e11664. https://doi.org/10.1002/sam.11664

**APPENDIX A. PROOF OF LEMMA 3**

In this section, we include results from an extra simulation study that showcases the finite-sample performance of our proposed method, SciDNet. We also provide a comparative analysis with other existing methods. The appendix demonstrates the phase transition property of feature importance through both synthetic and real data analysis. Additionally, we furnish details regarding model implementation, a link to the code, and an extensive discussion on hyperparameter settings.

**Lemma 3.** Sharp tail bound for weighted noncentral \( \chi^2 \) distribution.

Consider a weighted noncentral \( \chi^2 \) distributed random variable \( Y = \sum_{i=1}^{k} u_i Z_i^2, Z_i \sim N(\mu, 1) \) independently

\[ \mathbb{E}[Y] = \sum_{i=1}^{k} u_i \mu_i^2 + \sum_{i=1}^{k} \mu_i, \quad \text{Var}(Y) = \sum_{i=1}^{k} \mu_i^2 \]
and \( \sum_{i=1}^k \mu_i^2 = \lambda \). Then, for the centralized random variable \( X = Y - \sum_{i=1}^k u_i (1 + \mu_i^2) \), the following sharp tail bound holds: there exists constants \( c, c', C > 0 \), such that

\[
P(X \geq x) \leq c \exp\left(-\frac{ct^2}{2k + 2 \sum_{i=1}^k u_i \mu_i - k \sum_{i=1}^k u_i}\right),
\]

\( \forall 0 < x < c' \left( 2k + 2 \sum_{i=1}^k u_i \mu_i^2 - \sum_{i=1}^k u_i \right) \)

Proof. The moment-generating function of \( X \):

\[
\phi_X(t) = \exp\left[ \frac{2t^2}{1-2t} \sum_{i=1}^k u_i \mu_i - \frac{k}{2} (\log(1 - 2t) + 1) \right] + \left( k - \sum_{i=1}^k u_i \right).
\]

Note, for \( 0 \leq t \leq 1/2, 2t^2 \leq -\log(1 - 2t) - 2t \leq \frac{2t^2}{1-2t} \).

This implies, for \( 0 \leq t \leq 2/5 \),

\[
t^2 \left( 2k + 2 \sum_{i=1}^k u_i \mu_i^2 - k \sum_{i=1}^k u_i \right) \leq \log(\phi_X(t))
\]

\[ \leq 5t^2 \left( 2k + 2 \sum_{i=1}^k u_i \mu_i^2 - \sum_{i=1}^k u_i \right) \]

Next, the exact tail bound can be obtained by applying Theorem 1 from Zhang & Zhou [53]. □

APPENDIX B. ADDITIONAL TECHNICAL DETAILS

In this section, additional technical and implementation details on the proposed algorithm SciDNet are provided.

B.1. Hyperparameter selection

Recently developed deep learning (DL) models are generally governed by several hyperparameters and properly tuning them is necessary to get effective results. The proposed SciDNet relies on the following hyperparameters: (1) size of the active set \( \mathcal{S}_n \), (2) the intra-cluster correlation bound \( \rho \), (3) LassoNet tuning parameters \( \lambda \) and \( M \) and (4) \( \kappa \) used in neighborhood selection in cleaning step. SciDNet is fairly robust to most of the associated hyperparameters. We discuss a practical way to tune all these hyperparameters here:

1. To choose the size of the active set, we propose to select a bigger active set with size proportional to \( n/\log(n) \). As we further cluster the active variables in the clustering step, a slightly bigger active set with boost up the confidence of sure screening property, see the Appendix E for an example.

2. After clustering, the intra-cluster correlation bound \( \rho \) should be fixed at some higher value (usually at 0.9 or 0.95); otherwise, the cluster sizes will be inflated.

3. In the cleaning step, a thorough grid search has been done over \( \lambda \) considering \( \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_r \); in practice, a small value is fixed for \( \lambda_1 \) where all the variables are present in the model. Then the value of the tuning parameter gradually increased up to \( \lambda_r \), where there are no variables present in the model. The other hyperparameter for LassoNet is the hierarchy coefficient \( M \) for which we follow the path considered in Lemhadri et al. [28] and set \( M = 10 \). However, a more flexible approach would be a parallel grid search for \( M \) as well.

4. The neighborhood length \( \kappa \) can be chosen using the phase transition in the ranks of the importance scores, as described in Section 2.3.

B.2. Phase transition observed for the CCLE data

The main reason for the phase transition is that, for a null predictor \( X_j, j \in \mathcal{S}_0 \) different bootstrap replicates reshuffle its feature importance each time, whereas for a nonnull predictor \( X_j, j \in \mathcal{S}_0 \), the feature importance is much stable in different bootstrap replicates. SciDNet effectively captures this characteristic to identify the null features. As a demonstration, here we present in Figure B1 the bootstrap distribution of rank of the importance scores for the top 25 important cluster representatives via box plots. The green and purple colors respectively indicate if the cluster representatives are selected or rejected by the SciDNet. We can observe the phase transition consistently for all five drugs, and SciDNet selects only those important representatives with reduced variability over the bootstrap replicates.

APPENDIX C. MORE SIMULATION RESULTS

Here we demonstrate the finite sample performance of SciDNet under various linear and nonlinear models with varying multicollinearity levels under different signal-to-noise ratios.

C.1. Using Gaussian features

For the high-dimensional predictors, n i.i.d. copies are first generated from \( X \sim N_p(0, \Sigma) \), where \( n = 600 \), \( p = 5000 \), and the covariance matrix \( \Sigma \) is chosen as a Toeplitz matrix with \( \Sigma_{ij} = \rho^{i-j} \). The value of \( \rho \) is varied to explore different correlation strengths. We set the set of truly significant variables \( S = \{100,200,300,400,500\} \) with \( s = 5 \).

The response \( y \) is generated from \( y = g(x) + \epsilon \). Here we entertain the following three models:

1. **Linear**: \( g(x) = x_S \beta_S \) with \( \beta_S \) generated from \( N(2, 0.1) \) independently and \( \beta_S = 0 \),
2. Nonlinear additive: \( g(x) = 2x_{100} + 2x_{300}^3 + e^{x_{500}} + 6\sin(x_{400}) + 2\text{ReLU}(x_{500}^3) \), where \( \text{ReLU}(x) = \max(x, 0) \)

3. Nonlinear with interaction: \( g(x) = 2x_{100} + 2x_{200}^3 + e^{x_{200}} + 6x_{400}x_{500} \)

In each cases, the random noise \( \epsilon \) is independently generated from \( N(0, \sigma^2) \), where the value of \( \sigma^2 \) is chosen maintaining the signal-to-noise ratio at the desired level. To this end, we define the signal-to-noise ratio as \( \text{snr} = \frac{\text{var}(g(x))}{\sigma^2} \). Here we consider three levels of \( \text{snr} = 9 : 1, 8 : 2 \) and \( 7 : 3 \). Table S3 shows that SciDNet continues to maintain satisfactory power while successfully controlling the FDR below the threshold \( q = 0.01, 0.05, 0.1, 0.15 \). The average cluster size is observed at 8.3 for \( \rho = 0.9 \) and 13.4 for \( \rho = 0.95 \).

C.2. Using non-Gaussian features

To check SciDNet's performance under non-Gaussian setup, \( n \) i.i.d copies of high-dimensional feature vector \( X \) are generated from multivariate \( t_p(5) \) distribution considering same correlation structure as in the previous Appendix C.1, with \( n = 600, p = 5000 \). The remaining simulation setting is consistent with the previous Section 2.1. The performance of SciDNet is presented at Table S4 which is quite analogous to the results of Gaussian features.
APPENDIX D. PERFORMANCE OF EXISTING FEATURE SELECTION METHODS IN THE PRESENCE OF HIGH MULTICOLLINEARITY

In this section, we present a numerical illustration of the performance of several recently proposed nonlinear FDR-controlled feature selection algorithms. The predictors are first generated from $X_i \sim N_p(0, \Sigma), i = 1, 2, \ldots, n$, for multiple combination of $(n, p)$ and the covariance matrix $\Sigma$ is chosen as a Toeplitz matrix with $\Sigma_{ij} = \rho^{|i-j|}, \rho = 0.1, 0.5,$ and 0.9. Under simplistic setting, the response $y$ is generated from $y = x_S \beta_S + \epsilon$, $S = \{5, 10, \ldots, 50\}, |S| = 10$, with $\beta_S$ generated from $N(\beta_0, 0.1)$ independently and $\beta_0 = 0$. The random noise $\epsilon \sim N(0, 1)$. We focus on the Model-X knockoff [8], SurvNet [44], and DeepPINK [52]. For a more rigorous analysis, we consider two different versions of Model-X knockoff: (1) Model-X-Estimated, where the knockoffs are generated using an estimated multivariate Gaussian distribution, and (2) Model-X-True, where the knockoffs are generated using the true data-generating multivariate Gaussian distribution mentioned above. For the knockoff generation, we consider the equicorrelated construction using the R package knockoff: The Knockoff Filter for Controlled Variable Selection. To implement the SurvNet and DeepPINK, we use the codes mentioned in the respective papers Song & Li [44]; and Lu et al. [52]. We set $q = 0.20$ as the FDR-control threshold.

Figure D1 reveals several interesting characteristics. Both Model-X-Estimated and Model-X-True maintain the power-FDR balance under a low correlation setup. However, with higher multicollinearity, Model-X-Estimated fails to control the FDR below the specified threshold while the Model-X-True controls the FDR efficiently. This disparity indicates Model-X procedure induces inflation in false discoveries if the knockoffs are not generated properly under a “difficult” situation. As expected, the DL-based algorithms, such as SurvNet and DeepPINK, work much better in big-n-small-p and low correlation setups but typically fail in other cases, indicating their reduced effectiveness in ultrahigh-dimensional data with small sample sizes.

APPENDIX E. MODEL IMPLEMENTATION DETAILS AND SENSITIVITY ANALYSIS

In this section, we mention the implementation details of SciDNet that we consider for the simulation study and real data analysis. To select the size of the active set $\hat{S}_n$ in the screening step, in consistence with Xue & Liang [51], we set $|\hat{S}_n| = \left\lfloor \frac{2n}{\log(n)} \right\rfloor$ by selecting the predictors with the top $|\hat{S}_n|$ Henze–Zirkler test statistic $\hat{w}_z$, where $[z]$ denotes the integer part of $z$. In all our simulation scenarios, we set $r = 0.9$, the hyperparameter for intra-cluster correlation bound to further integrate highly correlated conditionally dependent clusters. In cleaning step, for LassoNet 100-dimensional one-hidden-layer feed-forward
neural network has been used; more detailed model architecture can be found at appendix in Lemhadri et al. [28]. For creating the compact neighborhood in the cleaning step, each time we choose the value of $\kappa$ utilizing the phase transition property mentioned in Section 2.2 of main manuscript. The feature selection performance of the SciDNet is demonstrated by calculating the average power and cFDR along with their standard error observed in 50 Monte Carlo replications. Each data set is randomly divided into train, validation and test with a 70-10-20 split. To assess the prediction performance, the test Mean Square Error (MSE) before and after the variable selection has been shown as part of the simulation study. For the prediction model, a 40-dimensional two-hidden-layer feed-forward neural network with ReLU and linear activation function is considered with Adam as optimizer. For the regression tree, we used the bagging for further stabilization, as mentioned in Breiman [4]. The number of leaves and nodes are chosen by minimizing the MSE on validation set.

To access the error bar for the sensitivity analysis, we generate typical data using the polynomial setup (Section 3 in the main manuscript, with $\beta = 2, \sigma^2 = 1$) and rerun SciDNet 50 times on the same data and set $q = 0.1$ as FDR-control threshold. The mean and standard deviations from these 50 replications are following: \textbf{power} = 0.99 (0.01), \textbf{observed FDR} = 0.03 (0.03), \textbf{test error by LassoNet} = 1.048 (0.101), and \textbf{test error by SciDNet + RT} = 0.710 (0.002).

APPENDIX F. IMPORTANT CLUSTERS OF GENE DISCOVERED BY SCIDNET

F.1 For CCLE dataset

Table S5 presents all the selected cluster of genes by SciDNet for the five anticancer drugs considered. The genes in a single cluster are mentioned in the “{}.” Previous research on this gene-expression data has revealed several genes as biologically associated with the corresponding drugs. SciDNet successfully discovers these genes as the top-most important gene associated with the drugs. In Table S5, the selected genes, which are confirmed by previous domain research, are highlighted and corresponding references are mentioned in the column 3.

F.2 For Riboflavin dataset

The Riboflavin production dataset contains much complicated correlation structure than the CCLE data; see Figure 1 in the main manuscript for a visual illustration. As a result, SciDNet has produced much larger cluster of genes compared with the cluster sizes from the CCLE dataset. For example, the average cluster size for CCLR and Riboflavin dataset is respectively 2.5 and 17.78. Table S6 shows the nine selected cluster of genes selected by SciDNet while the FDR is controlled at $q = 0.15$. Additionally, SciDNet discovered the gene YCIC_at as one of the expressive genes related to riboflavin production which was identified by Bühlmann et al. [7] as a causal gene in this context.